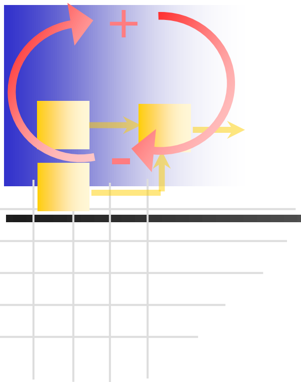
ESD.36 System Project Management

Lecture 5



Managing Iterations with DSM

Instructor(s)

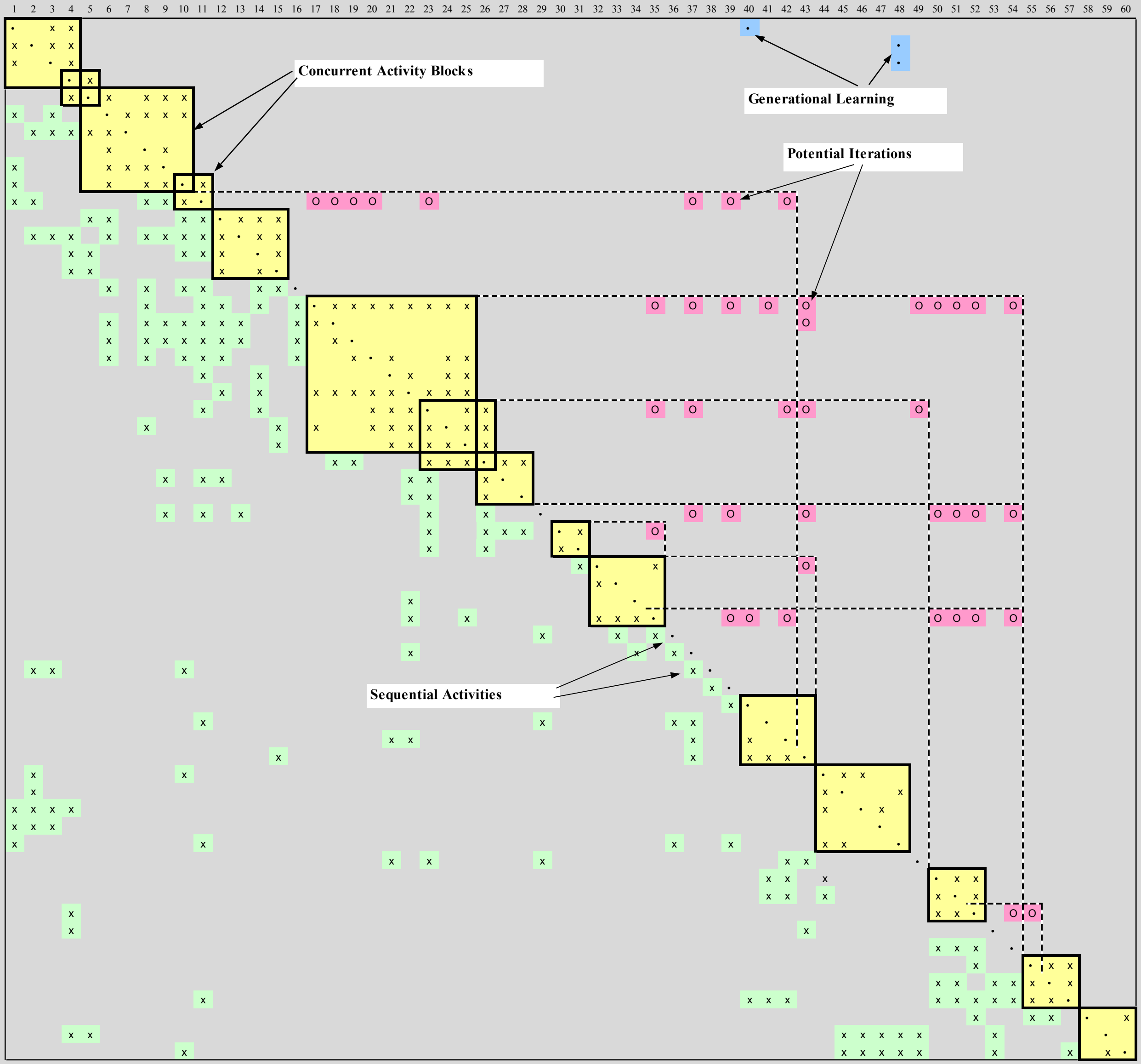
Prof. Olivier de Weck

Today’s Topics

- Iteration Models
  - Planned vs. Unplanned
  - Sequential vs. Parallel
- Product Development Process Analysis using DSM
  - Signal Flow Graph Method
  - Work Transformation Model
- Process/Project Improvement using DSM
  - Industrial Application of DSM at Ford
  - DSM use in Oil & Gas Projects at BP
- Introduce HW2

Semiconductor Development Example

1 Set customer target
2 Estimate sales volumes
3 Establish pricing direction
4 Schedule project timeline
5 Development methods
6 Macro targets/constraints
7 Financial analysis
8 Develop program map
9 Create initial QFD matrix
10 Set technical requirements
11 Write customer specification
12 High-level modeling
13 Write target specification
14 Develop test plan
15 Develop validation plan
16 Build base prototype
17 Functional modeling
18 Develop product modules
19 Lay out integration
20 Integration modeling
21 Random testing
22 Develop test parameters
23 Finalize schematics
24 Validation simulation
25 Reliability modeling
26 Complete product layout
27 Continuity verification
28 Design rule check
29 Design package
30 Generate masks
31 Verify masks in fab
32 Run wafers
33 Sort wafers
34 Create test programs
35 Debug products
36 Package products
37 Functionality testing
38 Send samples to customers
39 Feedback from customers
40 Verify sample functionality
41 Approve packaged products
42 Environmental validation
43 Complete product validation
44 Develop tech. publications
45 Develop service courses
46 Determine marketing name
47 Licensing strategy
48 Create demonstration
49 Confirm quality goals
50 Life testing
51 Infant mortality testing
52 Mfg. process stabilization
53 Develop field support plan
54 Thermal testing
55 Confirm process standards
56 Confirm package standards
57 Final certification
58 Volume production
59 Prepare distribution network
60 Deliver product to customers



x = Information Flows
🟨 = Planned Iterations
O = Unplanned Iterations
• = Generational Learning

Courtesy of Steve D. Eppinger. Used with permission.

Two Types of Iteration

Planned Iteration

- Caused by needs to "get it right the first time."
- We know where these iterations occur, but not necessarily how much.
- Planned iterations should be facilitated by good design methods, tools, and coordination.

Unplanned Iteration

- Caused by errors and/or unforeseen problems.
- We generally cannot predict which unplanned iterations will occur.
- Unplanned iterations should be minimized using risk management methods.

Design Iteration

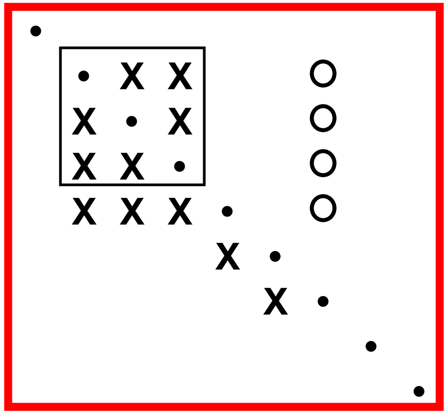
- Product development is fundamentally iterative — yet iterations are hidden.
- Iteration is the repetition of tasks due to the availability of new information.
  - changes in input information (upstream)
  - update of shared assumptions (concurrent)
  - discovery of errors (downstream)
- Engineering activities are repeated to improve product quality and/or to reduce cost.
- To understand and accelerate iterations requires
  - visibility of iterative information flows
  - understanding of the inherent process coupling

Discussion Point 1

- Have you experienced this distinction between planned and unplanned iterations in practice?
    - Are the boundaries blurred?
    - When have you iterated enough?

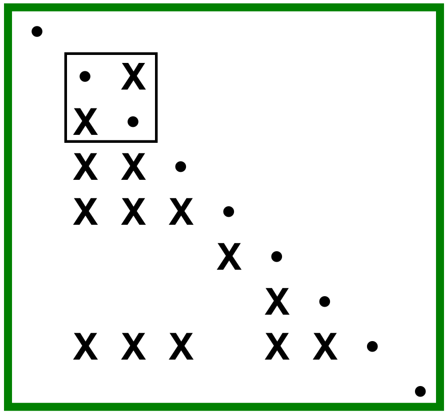
Instrument Cluster Development

Supplier A



	•								
Casing Design		•	X	X			O		
Wiring Layout		X	•	X			O		
Lighting Details		X	X	•			O		
Tooling		X	X	X	•		O		
Hard Prototype					X	•			
Testing						X	•		
								•	
									•

Supplier B



	•								
Casing Design		•	X						
Lighting Details		X	•						
Wiring Layout		X	X	•					
Soft Prototype		X	X	X	•				
Testing					X	•			
Revision						X	•		
Hard Tooling		X	X	X		X	X	•	
									•

Slower Design Process

Several planned iterations

Usually one unplanned iteration

Faster Design Process

Fewer planned iterations

Planned revision cycle

No unplanned iterations

Use of "Soft" Prototype

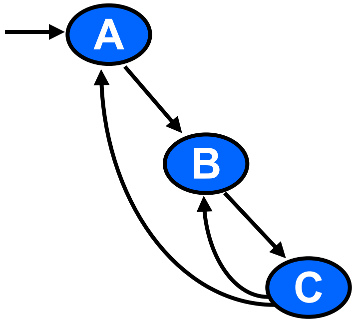
Two Iteration Styles

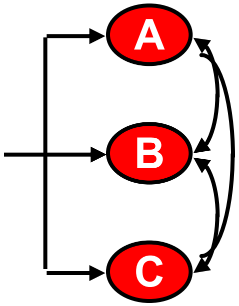
Sequential Iteration

- One activity is executed at a time.
- Models assume that probabilities determine the next actions.
- Signal Flow Graph Model

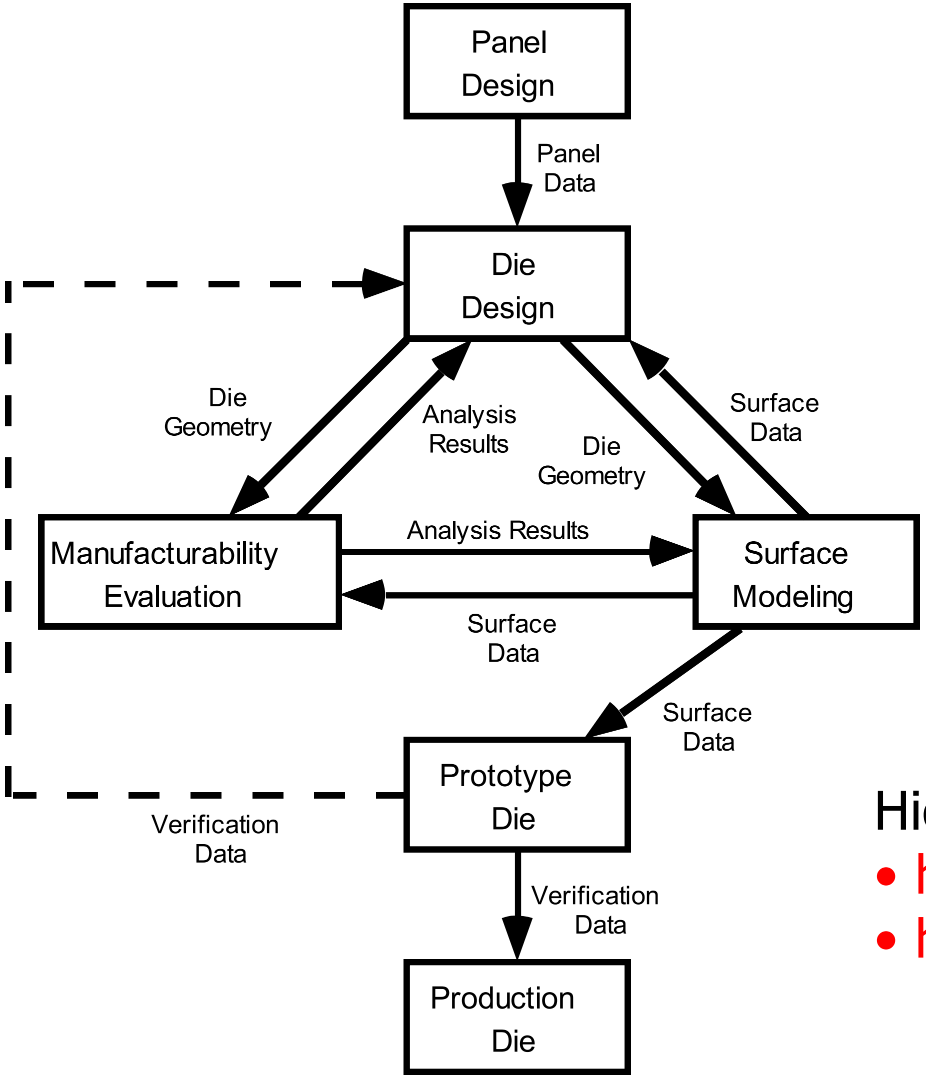
Parallel Iteration

- Several activities are executed at the same time.
- Models assume that rework is created for other coupled activities.
- Work Transformation Model





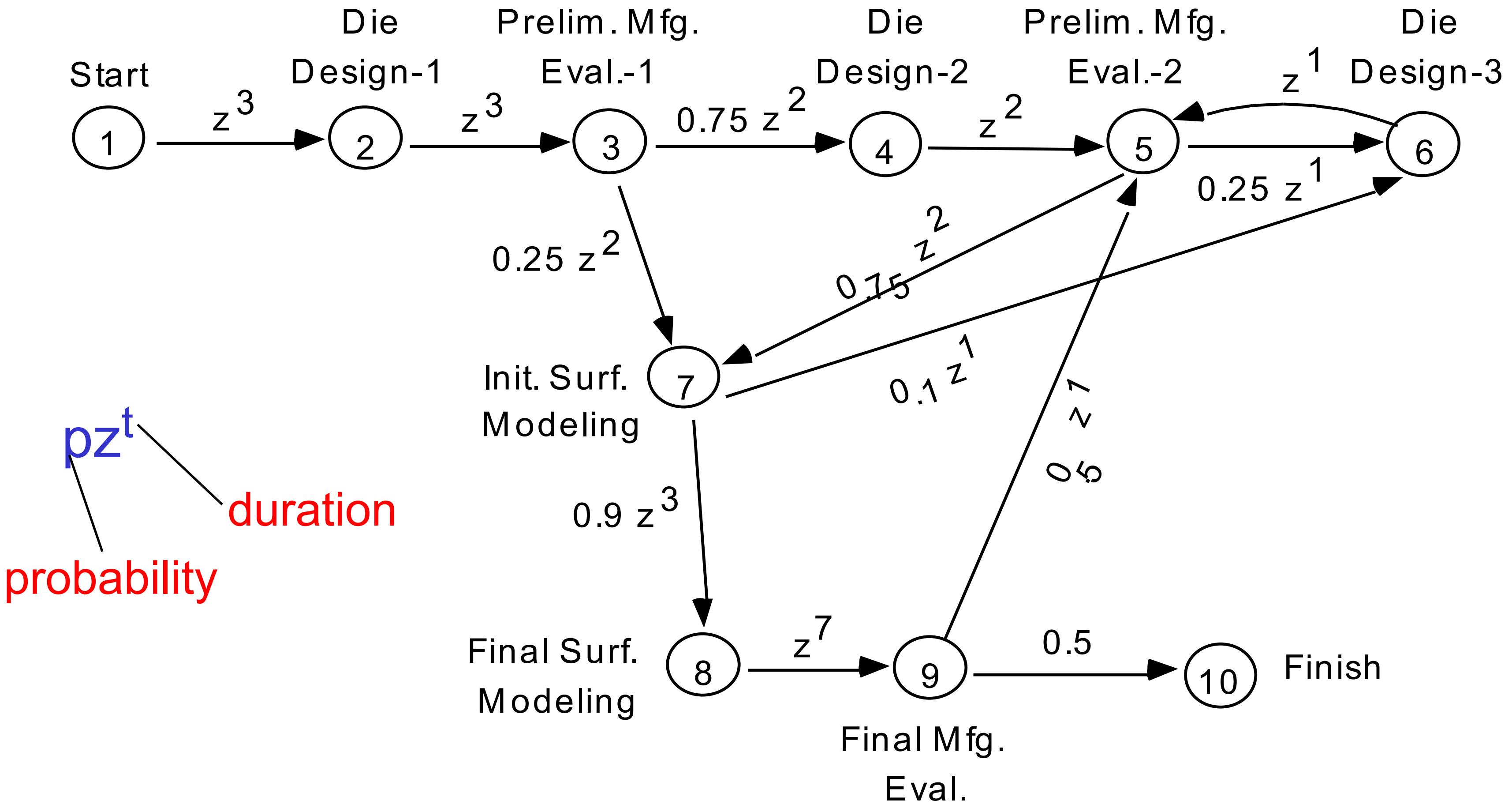
Stamping Die Development Process



Highly Iterative Process

- how often is each task carried out ?
- how long to complete?

Signal Flow Graph Model: Stamping Die Development



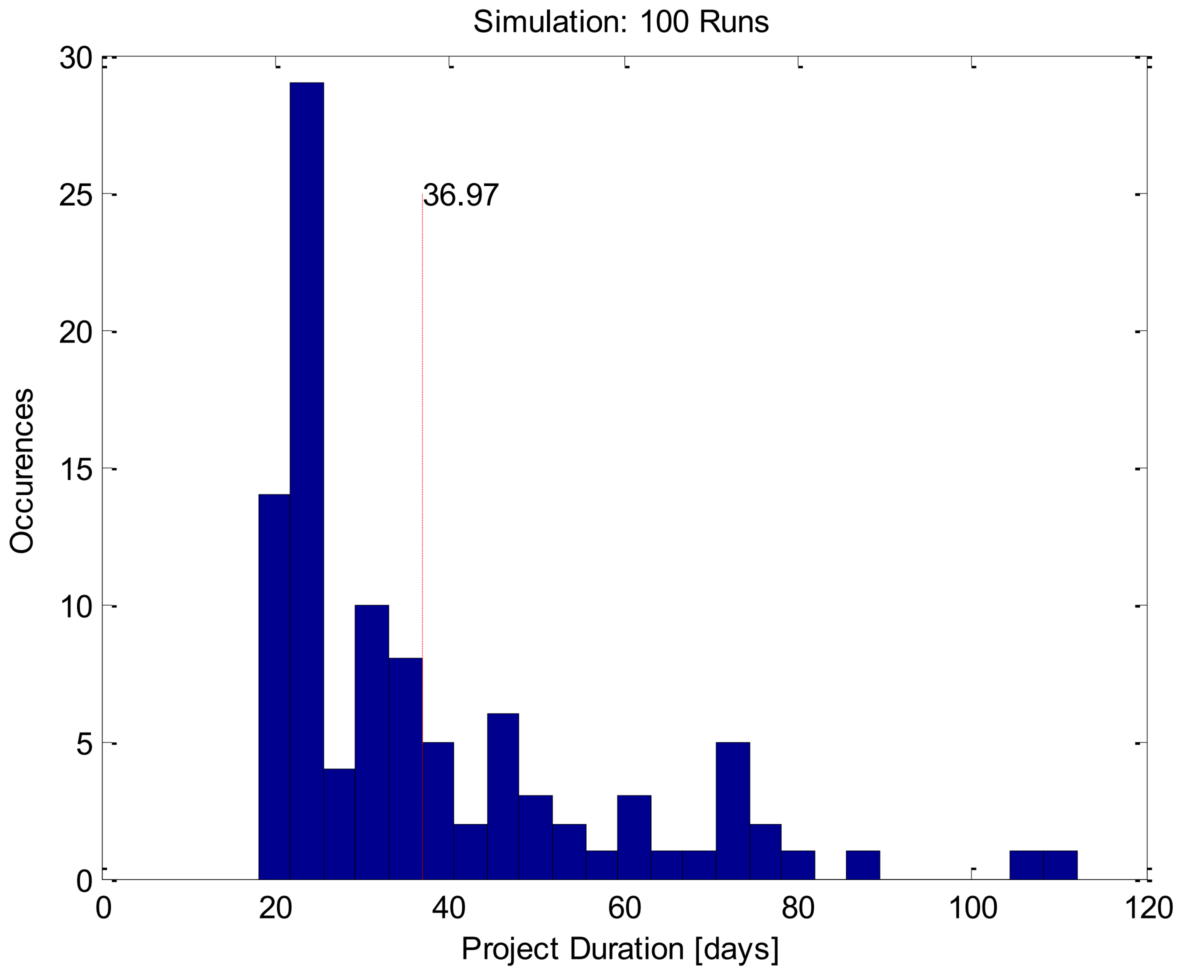
Matlab Simulation

- Review Signal Flow Simulation
  - State Transition Probability Matrix: P
  - State Transition Duration Matrix: T

- Implementation (die_sim.m)

```
while state<10
newstate= find(P(:,state));
cumprob= cumsum(P(newstate,state));
event=rand;
newind=max(find(event>[0 cumprob']));
% state transition
 time(ind)=time(ind)+T(newstate(newind),state);
state=newstate(newind);
end
```

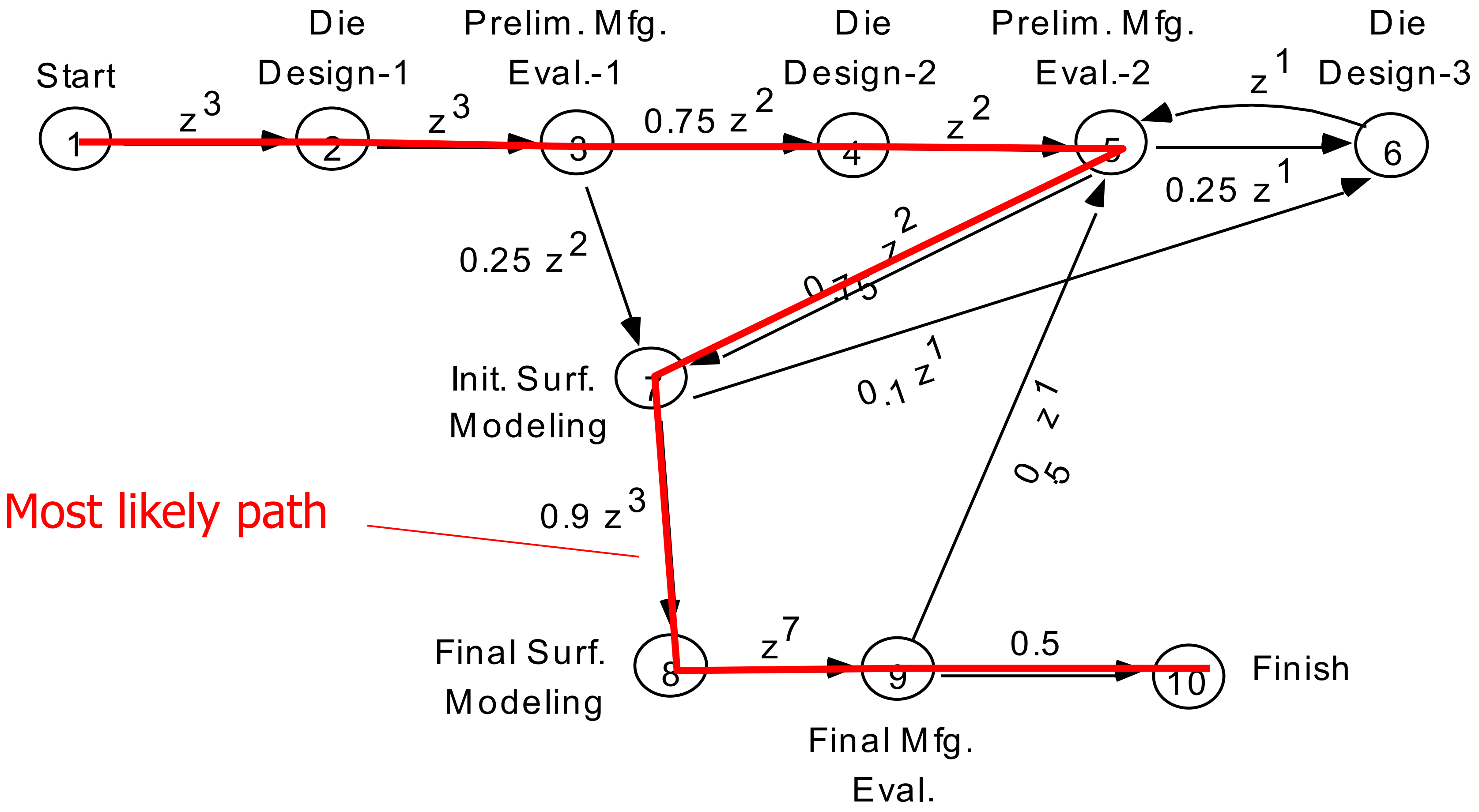
Computed Distribution of Die Development Timing



Estimate likely Completion time

What else can we do with the simulation?

Process Redesign/Refinement

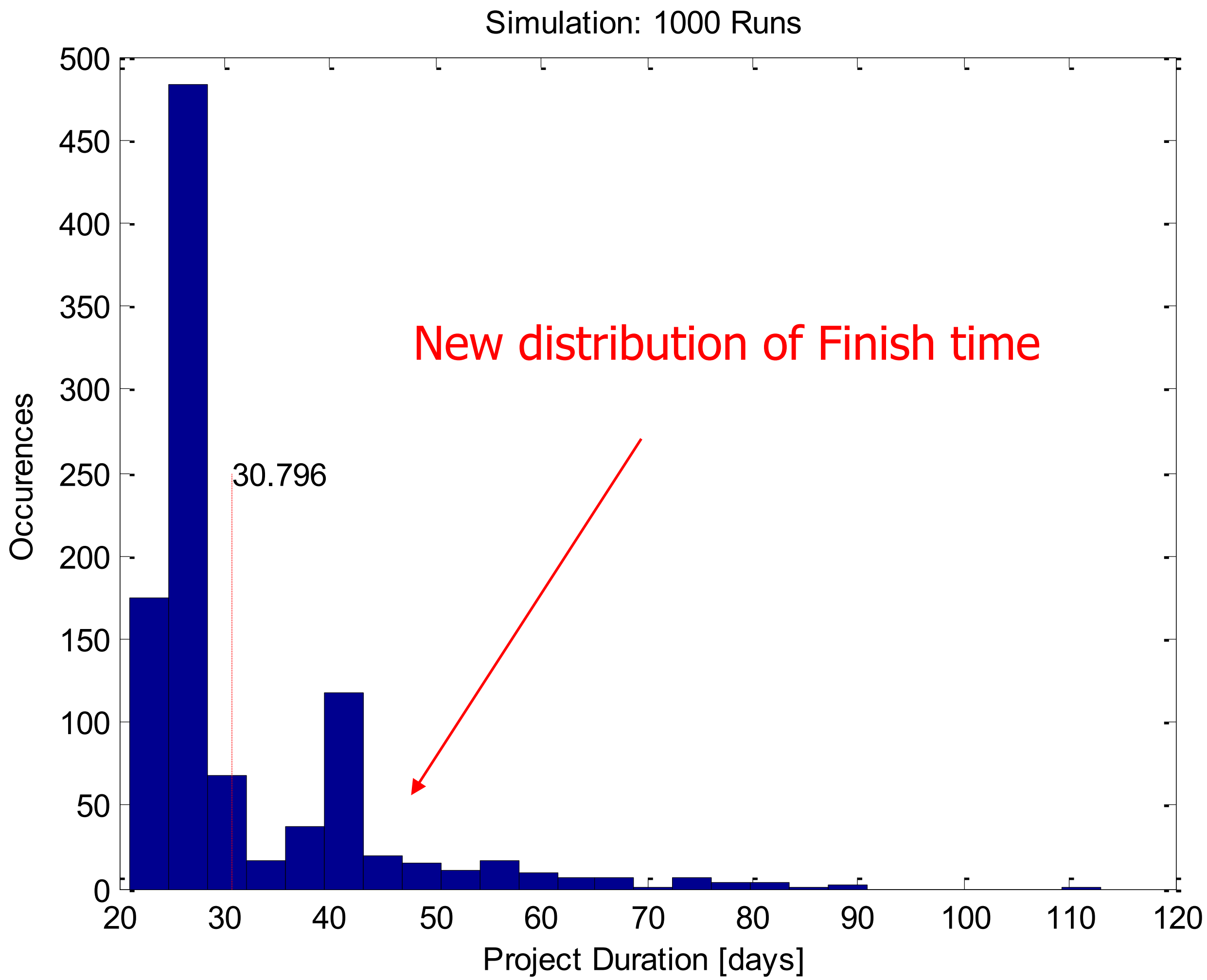


Most likely path

What-if analysis

- Spend more time on die design (1):
  - Increase time spent on initial die design (1) from 3 to 6 days
  - Increase likelihood of going to Initial Surface Modeling (7) from 0.25 to 0.75
    - Is this worthwhile doing?
    - Original E[F]=37 days
    - New E[F]= 37 days – no real effect ! Why?
- Spend more time on final surface modeling (8):
  - Increase time for that task from 7 to 10 days
  - Increase likelihood of Finishing from 0.5 to 0.75
    - New E[F] = 30.8 days
    - Why is this happening?

New Project Duration



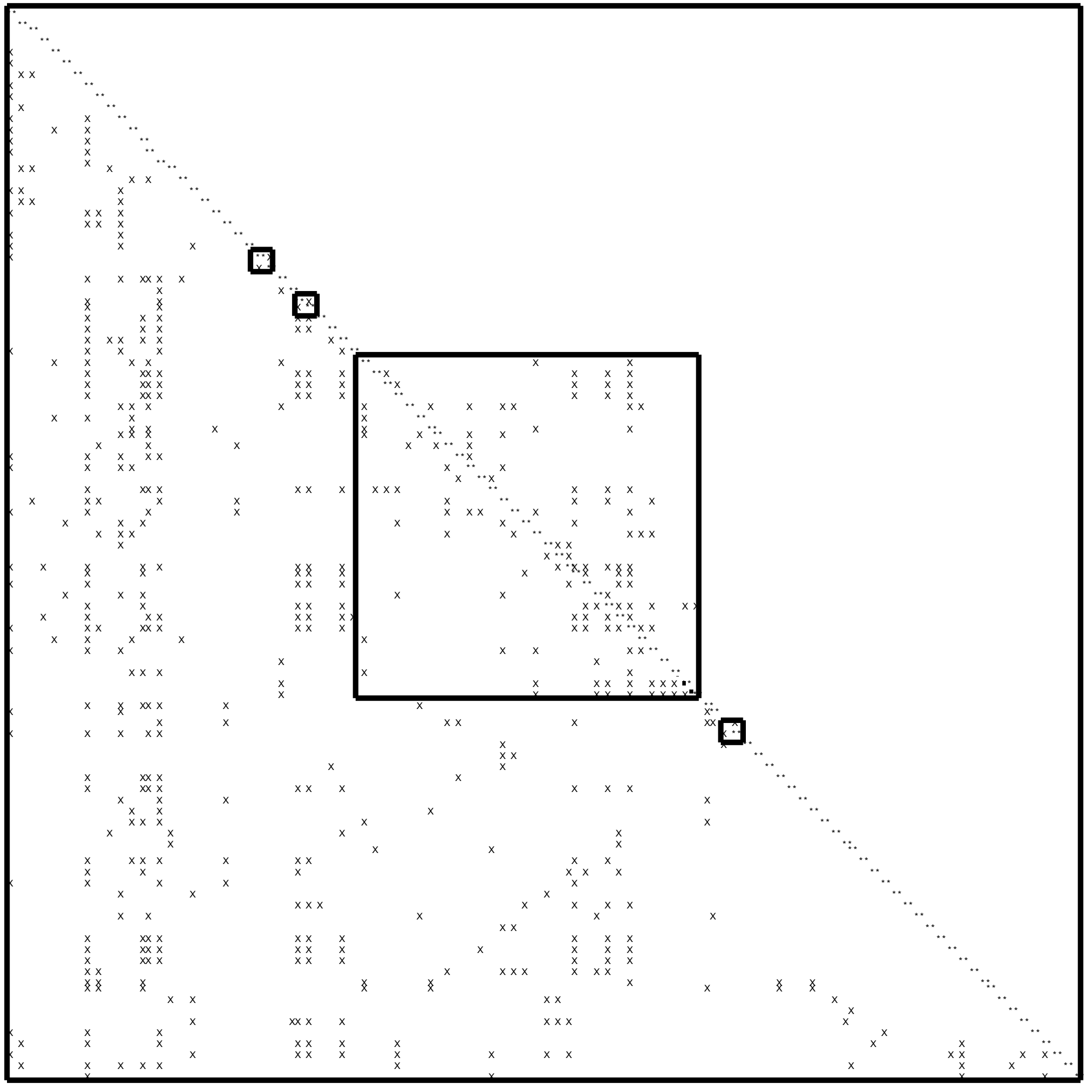
New distribution of Finish time

Concept Question 1

- In the Die Design Project, why did spending more time on final surface modeling (step 8) help reduce average completion time when spending more time on early die design (step 1) did not? Because …
  - The project avoids iterations altogether
  - The early die design cycle has been shortened by 20%
  - Fewer very long loops reduce the tail of the distribution
  - There is an increase in planned iterations which helps
  - It is a random result
  - I don’t know

Brake System Design Example

Work Transformation Model



105

parameters

Brake System Coupled Block

		33	34	35	37	40	44	45	46	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	04
33	Knuckle envelope & attach pts	**																•									•		
34	Pressure at rear wheel lock up		**	x																		X			•		•		
35	Brake torque vs. skidpoint			**	x																	x			x		x		
37	Line pressure vs. brake torque				**																	•			x		•		
40	Splash shield geometry—front	•				**		•				•			x	x											•	x	
44	Drum envelope & attach pts	•					**																						
45	Bearing envelope & attach pts	•						**										•									•		
46	Splash shield geometry—rear	•					•		**			•			•														
48	Air flow under car/wheel space					x			x	**		x																	
49	Wheel material										**	x																	
50	Wheel design									•		**			•														
51	Tire type/material										•		**	•															
52	Vehicle deceleration rate		X	X	X									**								•			x		•		
53	Temperature at components									x					**							•			x				•
54	Rotor cooling coeficient									x		x	•			**		x									x		
55	Lining—rear vol and area				x										•		**					x							
56	Rotor width									x						x		**									x	x	•
57	Pedal attach pts																		**	x	•								
58	Dash deflection																		x	**	x								
59	Pedal force (required)																			x	**	•	x		X	x	•		
60	Lining material—rear																•					**	•			x	•		
61	Pedal mechanical advantage																				x		**			x	•		
62	Lining—front vol & swept area				x										x									**	•				
63	Lining material—front																						x	•	**	X	x		x
64	Booster reaction ratio																					•	x		x	**	•		
65	Rotor diameter																					•	•		•	•	**	x	•
66	Rotor envelope & attach pts	x																										**	
104	Rotor material														x			•									•	•	**

•	Weak
x	Medium
X	Strong

The Work Transformation Model (Parallel Iteration Model)

$$u_{t+1} = Au_t$$

work vector

work transformation matrix

Assumptions

- All coupled tasks are attempted simultaneously.
- Off-diagonal elements correspond to fractions of each task's work which must be repeated during subsequent iterations.
- Objective is to characterize the nature of design iteration.

Work Transformation Model Mathematics

$$u_{t+1} = Au_t$$

work transformation equation

$$U = \sum_{t=0}^{\infty} u_t = (\sum_{t=0}^{\infty} A^t)u_0$$

total work vector

$$A = S\Lambda S^{-1}$$

eigenvalue decomposition

$$U = S\left(\sum_{t=0}^{\infty}\Lambda^t\right)S^{-1}u_0$$

substitution

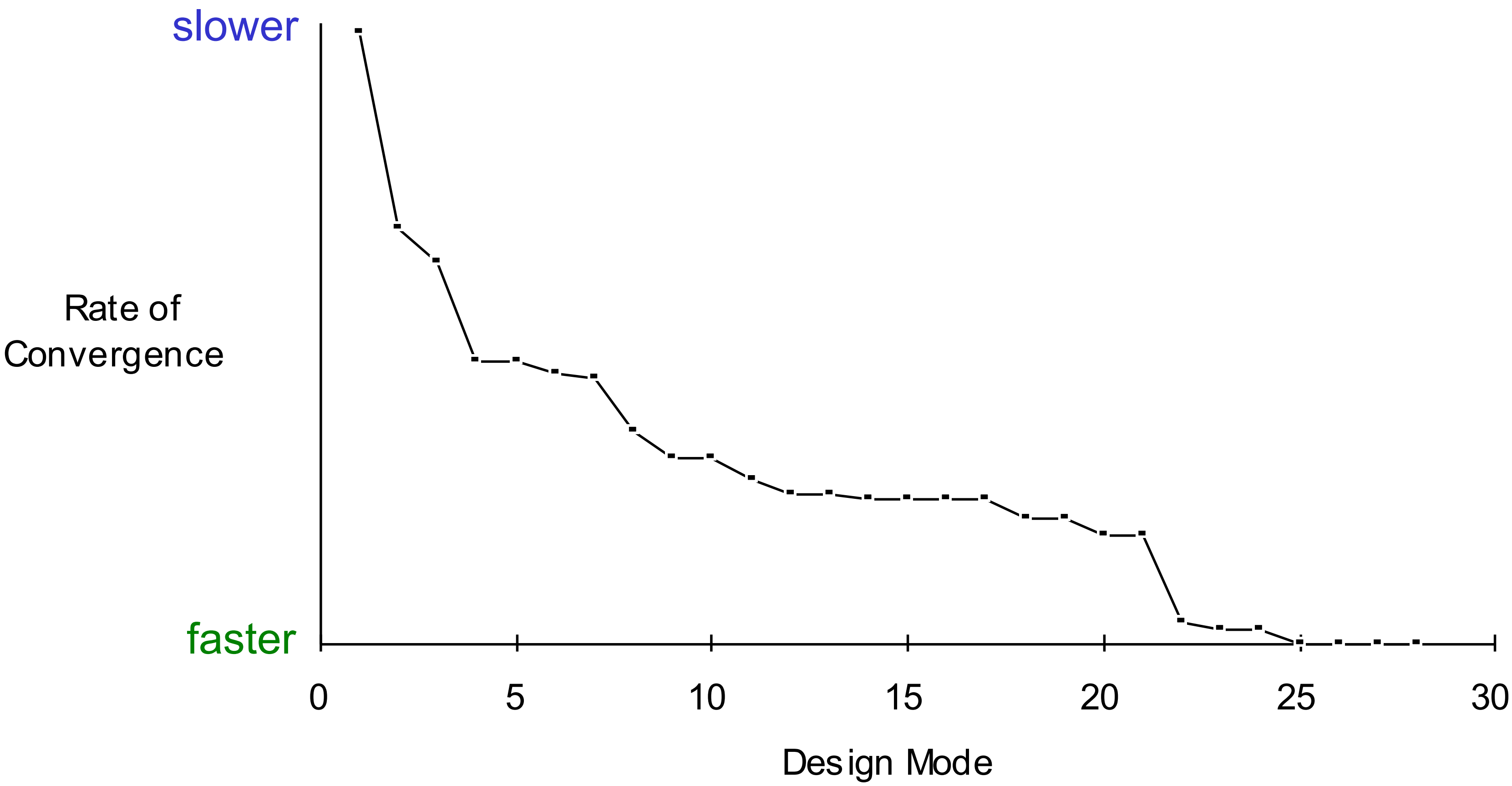
$$\left(\sum_{t=0}^{\infty}\Lambda^t\right) = (I-\Lambda)^{-1}$$

diagonal matrix of $\frac{1}{1-\lambda}$ terms

$$\underset{\text{total work}}{U} = \underset{\text{eigenvector matrix}}{S}\underbrace{\left[(I-\Lambda)^{-1}S^{-1}u_0\right]}_{\text{scaling vector}}$$

Total work is a scaling of the eigenvectors.

Brake System "Design Modes"



Brake System "Design Modes"

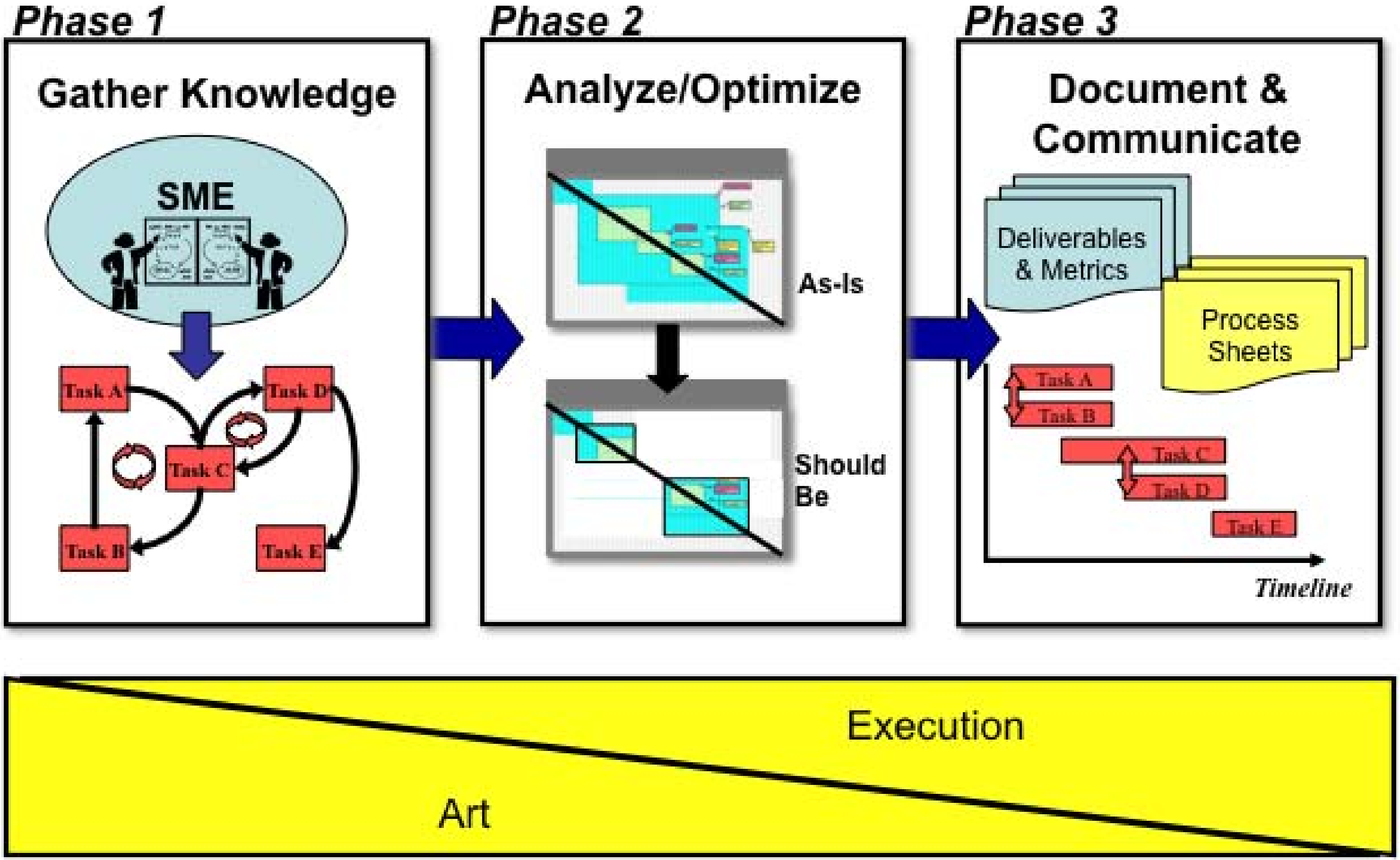
	First	Second
Knuckle envelope & attach pts	0.0157	0.1215
Pressure at rear wheel lock up	**0.4808**	0.0075
Brake torque vs. skidpoint	**0.4254**	0.0435
Line pressure vs. brake torque	0.1979	0.0228
Splash shield geometry—front	0.1109	**0.8328**
Drum envelope & attach pts	0.0011	0.0141
Bearing envelope & attach pts	0.0168	0.1356
Splash shield geometry—rear	0.0143	0.0654
Air flow under car/wheel space	0.0512	**0.5824**
Wheel material	0.0057	0.0610
Wheel design	0.0156	0.1051
Tire type/material	0.0731	0.0177
Vehicle deceleration rate	**1.0000**	0.0910
Temperature at components	0.1641	**0.3224**
Rotor cooling coefficient	0.1035	**0.9598**
Lining—rear vol and area	0.1479	0.0166
Rotor width	0.1043	**1.0000**
Pedal attach pts	0.1843	0.1584
Dash deflection	**0.3510**	0.2265
Pedal force (required)	**0.7818**	0.2317
Lining material—rear	0.1765	0.0587
Pedal mechanical advantage	**0.4193**	0.1749
Lining—front vol & swept area	0.1669	0.2052
Lining material—front	**0.4870**	0.0417
Booster reaction ratio	**0.3502**	0.0787
Rotor diameter	0.1117	0.0463
Rotor envelope & attach pts	0.0057	0.0705
Rotor material	0.0757	**0.3168**

Stopping Performance Design Mode

Thermal Design Mode

Application of DSM at Ford

Three-Phased Approach



Phase 1

Gather Knowledge

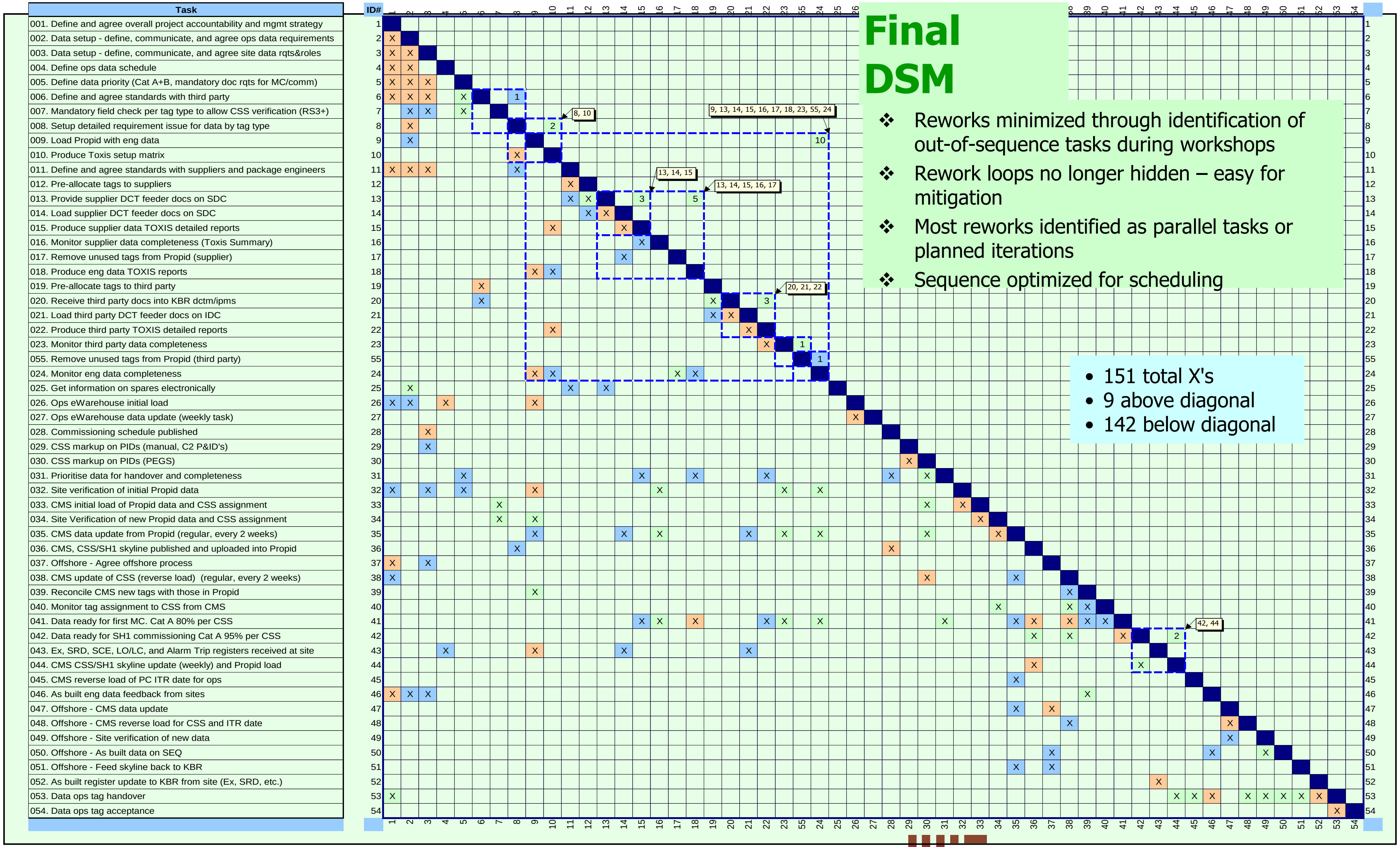
Phase 2

Analyze/Optimize

Phase 3

Document & Communicate

Application of DSM at BP



Task	ID#
001. Define and agree overall project accountability and mgmt strategy	1
002. Data setup - define, communicate, and agree ops data requirements	2
003. Data setup - define, communicate, and agree site data rqts&roles	3
004. Define ops data schedule	4
005. Define data priority (Cat A+B, mandatory doc rqts for MC/comm)	5
006. Define and agree standards with third party	6
007. Mandatory field check per tag type to allow CSS verification (RS3+)	7
008. Setup detailed requirement issue for data by tag type	8
009. Load Propid with eng data	9
010. Produce Toxis setup matrix	10
011. Define and agree standards with suppliers and package engineers	11
012. Pre-allocate tags to suppliers	12
013. Provide supplier DCT feeder docs on SDC	13
014. Load supplier DCT feeder docs on SDC	14
015. Produce supplier data TOXIS detailed reports	15
016. Monitor supplier data completeness (Toxis Summary)	16
017. Remove unused tags from Propid (supplier)	17
018. Produce eng data TOXIS reports	18
019. Pre-allocate tags to third party	19
020. Receive third party docs into KBR dctm/ipms	20
021. Load third party DCT feeder docs on IDC	21
022. Produce third party TOXIS detailed reports	22
023. Monitor third party data completeness	23
055. Remove unused tags from Propid (third party)	55
024. Monitor eng data completeness	24
025. Get information on spares electronically	25
026. Ops eWarehouse initial load	26
027. Ops eWarehouse data update (weekly task)	27
028. Commissioning schedule published	28
029. CSS markup on PIDs (manual, C2 P&ID's)	29
030. CSS markup on PIDs (PEGS)	30
031. Prioritise data for handover and completeness	31
032. Site verification of initial Propid data	32
033. CMS initial load of Propid data and CSS assignment	33
034. Site Verification of new Propid data and CSS assignment	34
035. CMS data update from Propid (regular, every 2 weeks)	35
036. CMS, CSS/SH1 skyline published and uploaded into Propid	36
037. Offshore - Agree offshore process	37
038. CMS update of CSS (reverse load) (regular, every 2 weeks)	38
039. Reconcile CMS new tags with those in Propid	39
040. Monitor tag assignment to CSS from CMS	40
041. Data ready for first MC. Cat A 80% per CSS	41
042. Data ready for SH1 commissioning Cat A 95% per CSS	42
043. Ex, SRD, SCE, LO/LC, and Alarm Trip registers received at site	43
044. CMS CSS/SH1 skyline update (weekly) and Propid load	44
045. CMS reverse load of PC ITR date for ops	45
046. As built eng data feedback from sites	46
047. Offshore - CMS data update	47
048. Offshore - CMS reverse load for CSS and ITR date	48
049. Offshore - Site verification of new data	49
050. Offshore - As built data on SEQ	50
051. Offshore - Feed skyline back to KBR	51
052. As built register update to KBR from site (Ex, SRD, etc.)	52
053. Data ops tag handover	53
054. Data ops tag acceptance	54

Final DSM

- ❖ Reworks minimized through identification of out-of-sequence tasks during workshops
- ❖ Rework loops no longer hidden – easy for mitigation
- ❖ Most reworks identified as parallel tasks or planned iterations
- ❖ Sequence optimized for scheduling

- 151 total X's
- 9 above diagonal
- 142 below diagonal

Concept Question 2

- What determines the "optimal" balance of planned versus unplanned (rework) iterations in projects
  - A) Novelty and complexity of work to be done
  - B) Schedule pressure to complete
  - C) Software versus hardware content
  - D) A and B
  - E) B and C
  - F) All of the above

Summary: Iterations

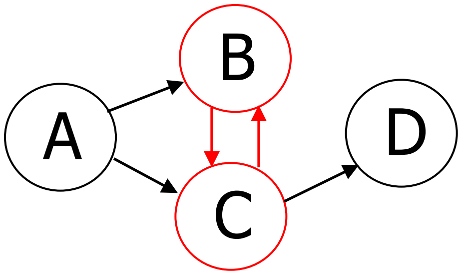
- Development projects are inherently iterative.
- An understanding of the coupling is essential.
- Iteration results in improved quality.
- Iteration can be accelerated through:
  - information technology (faster iterations)
  - coordination techniques (faster iterations)
  - decreased coupling (fewer iterations) → modular design?
- There are two fundamental types of iteration:
  - planned iterations (getting it right the first time)
  - unplanned iterations (fixing it when it's not right)
- Mature processes have more planned and fewer unplanned iterations.

Always ask as a project manager: Where do we expect iterations?

Discover Loops (Length 2)

- Turn DSM into a binary matrix
  - Replace “X” and “ “ with 1 and 0
  - Square binary matrix
  - Find non-zero diagonals

Example:



A			
X	B	X	
X	X	C	
		X	D

{0,1}

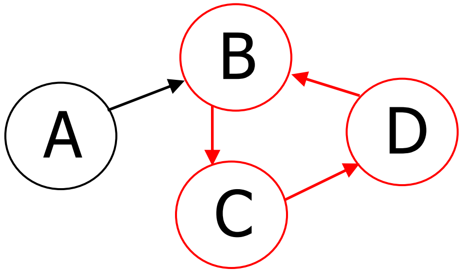
0	0	0	0
1	0	1	0
1	1	0	0
0	0	1	0

^2

0	0	0	0
1	1	0	0
1	0	1	0
1	1	0	0

Discover Loops (Length 3)

Example:



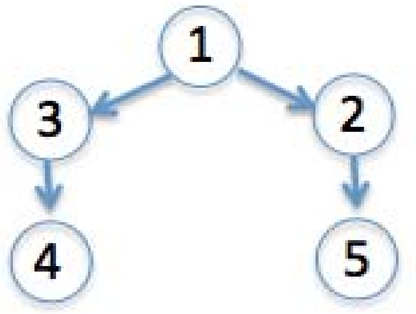
A			
X	B		X
	X	C	
		X	D

0	0	0	0
1	0	0	1
0	1	0	0
0	0	1	0

^3 ⇨

0	0	0	0
0	1	0	0
0	0	1	0
1	0	0	1

Visibility Matrix



$$V = \sum_{n=1}^{4} A^n = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad A^2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad A^3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Visibility Matrix is a way to find loops and most influential tasks

HW2: DSM Model of CityCar

- Still you have the CityCar Project Manager Role
- Translate CPM → DSM
  - Network Graph → Matrix
- Add Iterations
- Find Loops
- Reorganize DSM
  - Sequence (reorder tasks)
  - Partition (cluster coupled tasks)
  - Tearing (break loops)

ESD.36 System Project Management
Fall 2012