

***Lean Aircraft Initiative
Plenary Workshop***

***Cycle Time Reduction Using
Design Structure Matrices***



October 8 -9, 1997

**Presented By:
Tyson R. Browning
Product Development Focus Group**

- Some cycle time reduction challenges
- Task-based *design structure matrices* (DSMs)
- How DSM method meets challenges
- Part 2: David Grose on applications at Boeing

Perspective:



Some Cycle Time Reduction Challenges

- **Inefficient personnel and resource distribution**
 - People
 - Facilities & tools
 - Information
- **Unstable requirements**
- **Long and varied activity pipelines**
- **Lack of activity coordination**
- **Highly coupled activities**
- **Delayed decisions**
- **Rework**
- **Schedule too ambitious to begin with**
- **Hoped for solutions have adverse side effects**

- **Account for many aspects of schedule overrun**
- **Design rework, change, refinement**
- **Difficult to plan for**
- **Intentional iterations**
 - Improve design quality
 - Reduce performance risk
- **Unintentional iterations caused by**
 - Poor task sequencing quality
 - Poor communication and coordination quality
 - Design mistakes
 - Poor requirements quality, simplicity, stability
- **More difficult for complex system development**

- **One approach: overlapping tasks**
 - Can cause iteration if tasks are coupled
- **True cycle time reduction requires**
 - Systems view
 - Much information processing for complex relationships
- **Need for method**
 - to structure complex interactions
 - to rework sequences of tasks
- **A better approach: Design Structure Matrix**
 - To manage task couplings and iterations
- **To reduce schedule overrun risk**
- **To reduce cycle time**

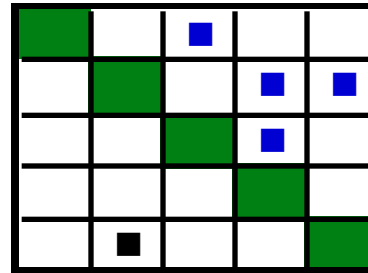
Design Structure Matrices

- **Origins**
 - Solving systems of equations (1960's)
 - Network precedence diagrams (1960's)
 - Matrix mathematics (1960's and 1970's; Warfield, et al.)
 - Similar to N² diagrams in systems engineering
- **Formal definition**
 - Design processes (1981, Steward)
 - Multiple applications in automotive, electronics, and semiconductor industries (early 1990's; Eppinger, et al.)
- **Compared with PERT**
 - Shows task precedence and coupling
 - Explicit notation of iteration
 - Simpler representation of complex process

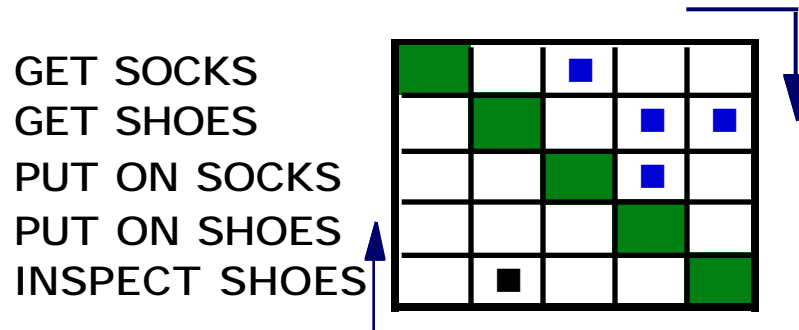
**LEAN AIRCRAFT
INITIATIVE**

Introduction to Task-based or Schedule DSMs

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



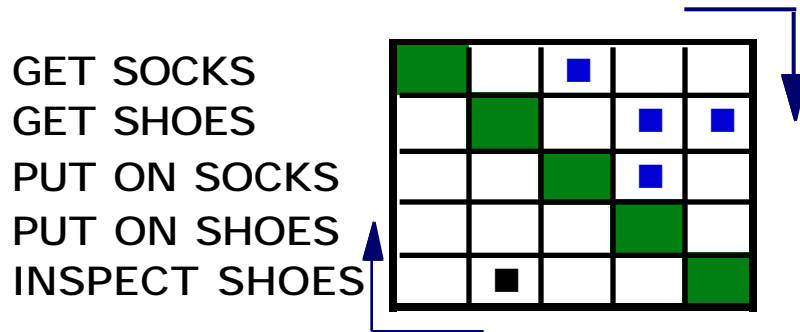
Diagrams adapted from Denker, Stephen. "ANew Way to Think About Problems" Presentation to PMI, 6/19/97.



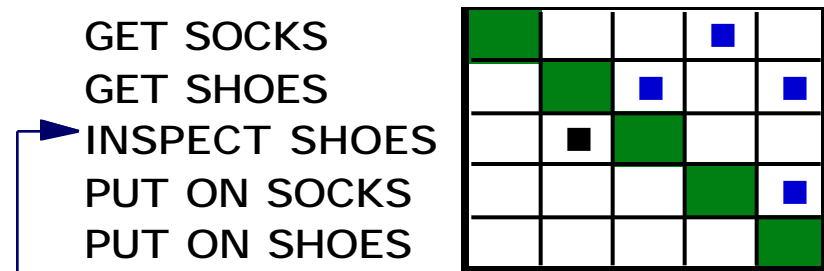
- ◆ **Precedence diagram: information dependencies**
- ◆ **Sequence tasks**
- ◆ **First goal: reduce iterations and their scope**
- ◆ **Problem: coupled tasks**
- ◆ **Identify parallel tasks**

Diagrams adapted from Denker, Stephen. "ANew Way to Think About Problems" Presentation to PMI, 6/19/97.

Introduction to Task-based or Schedule DSMs



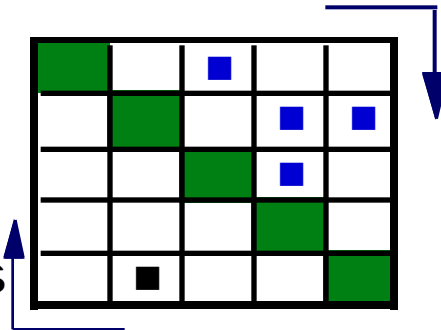
- ◆ **Precedence diagram: information dependencies**
- ◆ **Sequence tasks**
- ◆ **First goal: reduce iterations and their scope**
- ◆ **Problem: coupled tasks**
- ◆ **Identify parallel tasks**



Diagrams adapted from Denker, Stephen. "ANew Way to Think About Problems" Presentation to PMI, 6/19/97.

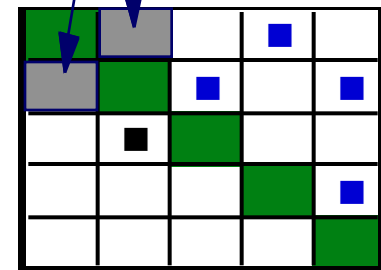
Introduction to Task-based or Schedule DSMs

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



No interactions between
parallel tasks

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



- ◆ Precedence diagram: information dependencies
- ◆ Sequence tasks
- ◆ First goal: reduce iterations and their scope
- ◆ Problem: coupled tasks
- ◆ Identify parallel tasks

Diagrams adapted from Denker, Stephen. "ANew Way to Think About Problems" Presentation to PMI, 6/19/97.

DSMs Show Opportunities to Reduce Cycle Time

- **DSM: a *description* of the current process**
- **Modify DSM to achieve a *prescription* for a process with reduced risk and cycle time**
- **Capabilities**
 - **More accurately manage schedule**
 - **Systems view of many cycle time drivers**
 - **Quickly examine potential task sequence changes**
 - **Intelligently place tasks in parallel**
 - **Deploy resources to reduce unintentional iterations**

Kodak “Cheetah” Project

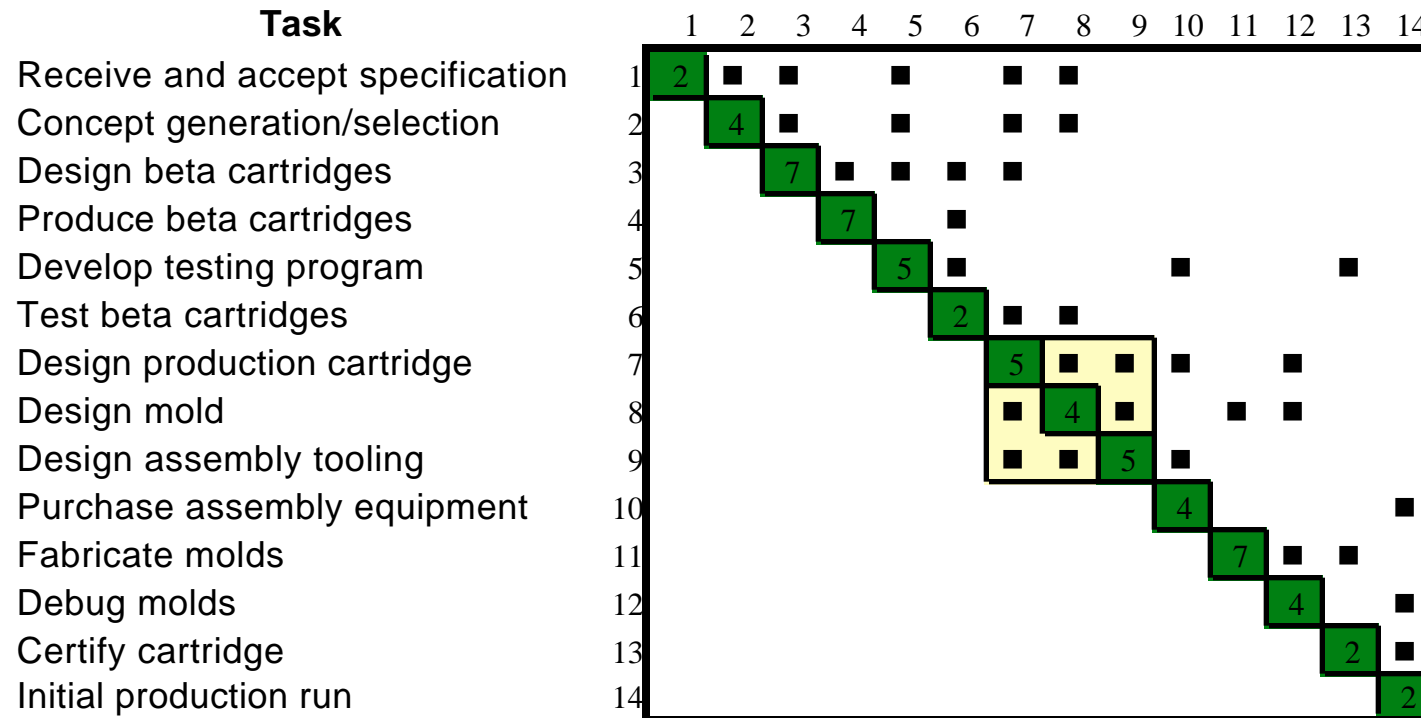


Diagram adapted from Ulrich and Eppinger, *Product Design and Development*, McGraw-Hill, 1995, p. 262

Kodak “Cheetah” Project

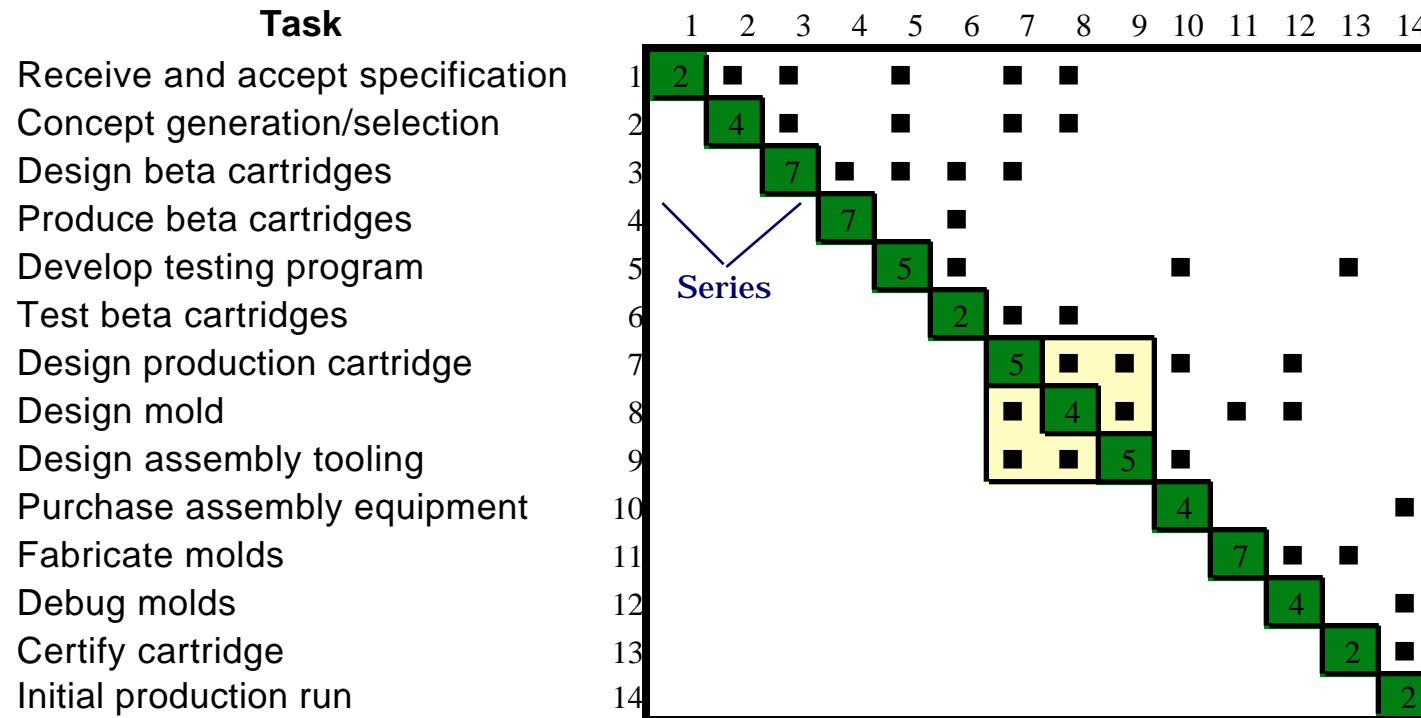


Diagram adapted from Ulrich and Eppinger, *Product Design and Development*, McGraw-Hill, 1995, p. 262

Kodak “Cheetah” Project

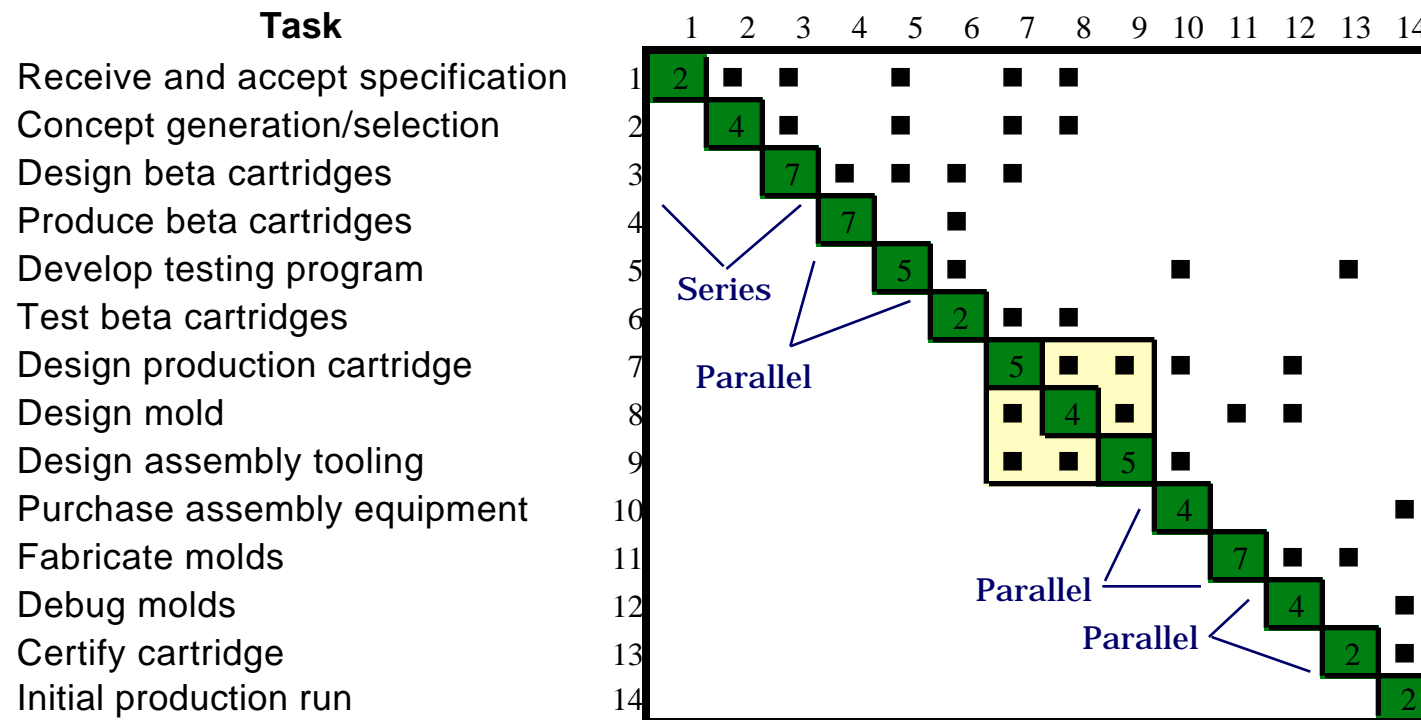


Diagram adapted from Ulrich and Eppinger, *Product Design and Development*, McGraw-Hill, 1995, p. 262

Kodak “Cheetah” Project

Task
Receive and accept specification
Concept generation/selection
Design beta cartridges
Produce beta cartridges
Develop testing program
Test beta cartridges
Design production cartridge
Design mold
Design assembly tooling
Purchase assembly equipment
Fabricate molds
Debug molds
Certify cartridge
Initial production run

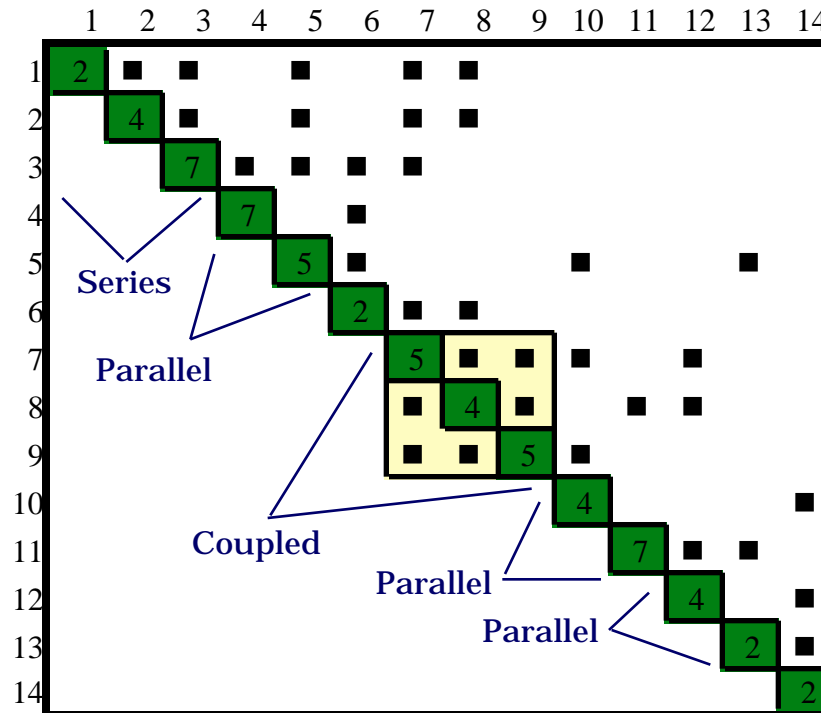
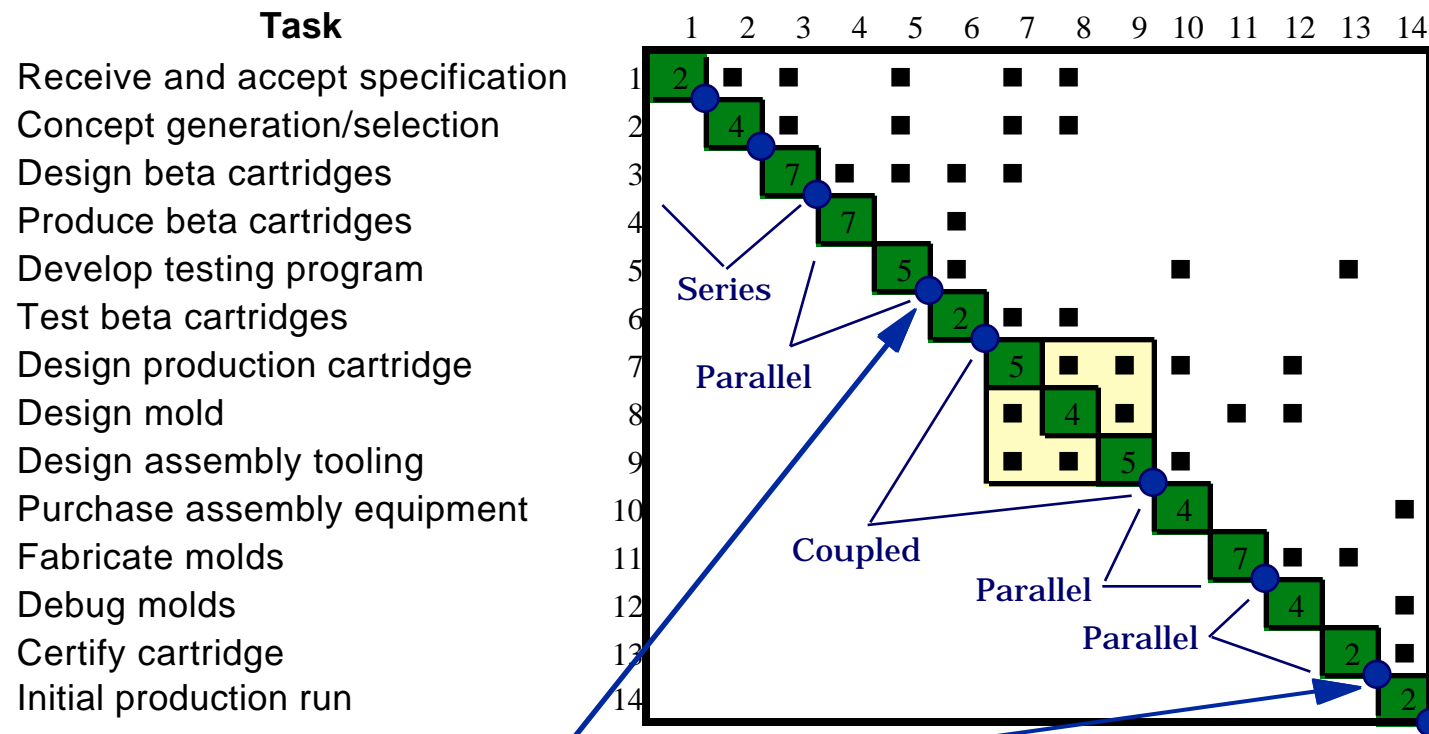


Diagram adapted from Ulrich and Eppinger, *Product Design and Development*, McGraw-Hill, 1995, p. 262

Kodak “Cheetah” Project



“Toll Gates” for Review

Diagram adapted from Ulrich and Eppinger, *Product Design and Development*, McGraw-Hill, 1995, p. 262

Kodak “Cheetah” Project



- Task**
- 1 Receive and accept specification
 - 2 Concept generation/selection
 - 3 Design beta cartridges
 - 4 Produce beta cartridges
 - 5 Develop testing program
 - 6 Test beta cartridges
 - 7 Design production cartridge
 - 8 Design mold
 - 9 Design assembly tooling
 - 10 Purchase assembly equipment
 - 11 Fabricate molds
 - 12 Debug molds
 - 13 Certify cartridge
 - 14 Initial production run

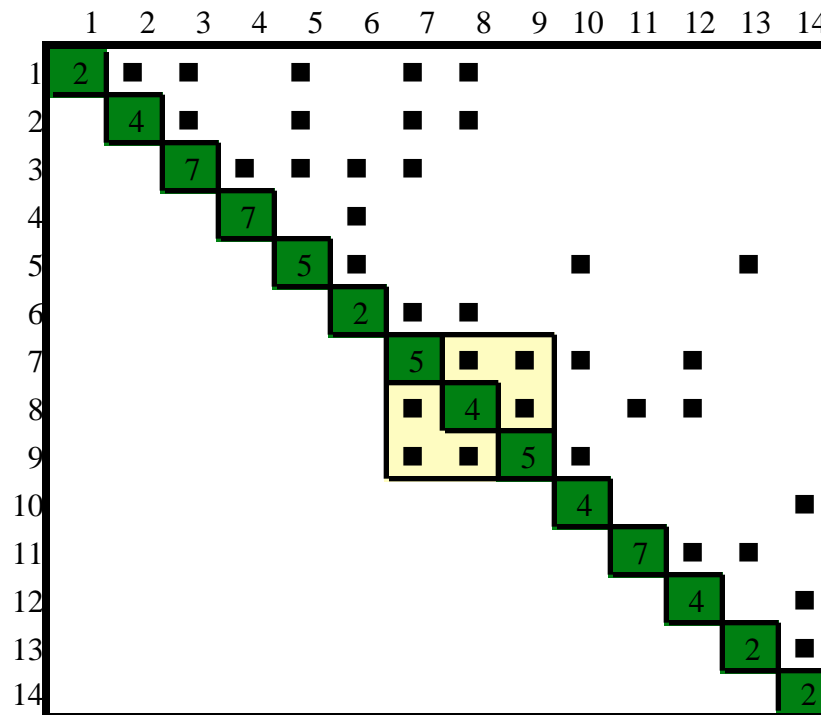


Diagram adapted from Ulrich and Eppinger, *Product Design and Development*, McGraw-Hill, 1995, p. 262

Kodak “Cheetah” Project

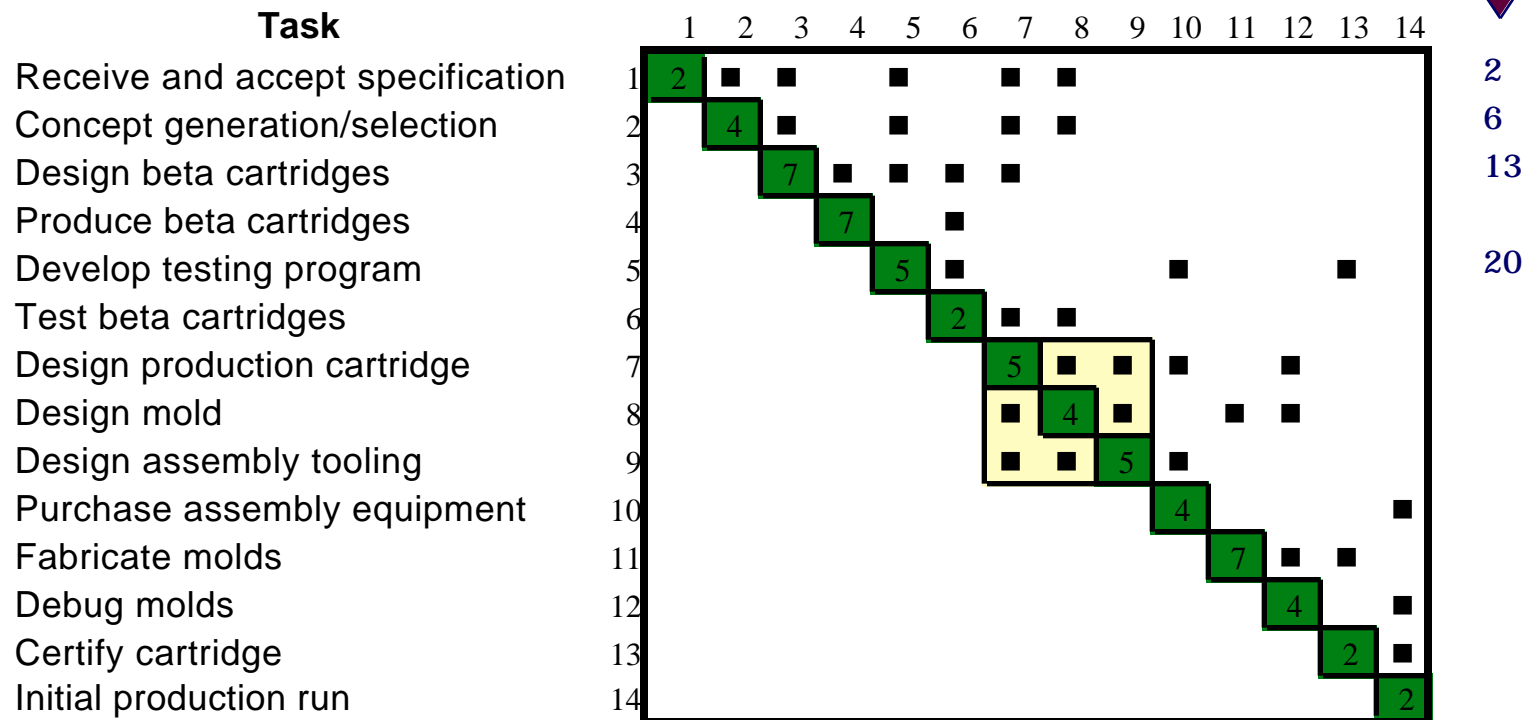


Diagram adapted from Ulrich and Eppinger, *Product Design and Development*, McGraw-Hill, 1995, p. 262

Kodak “Cheetah” Project

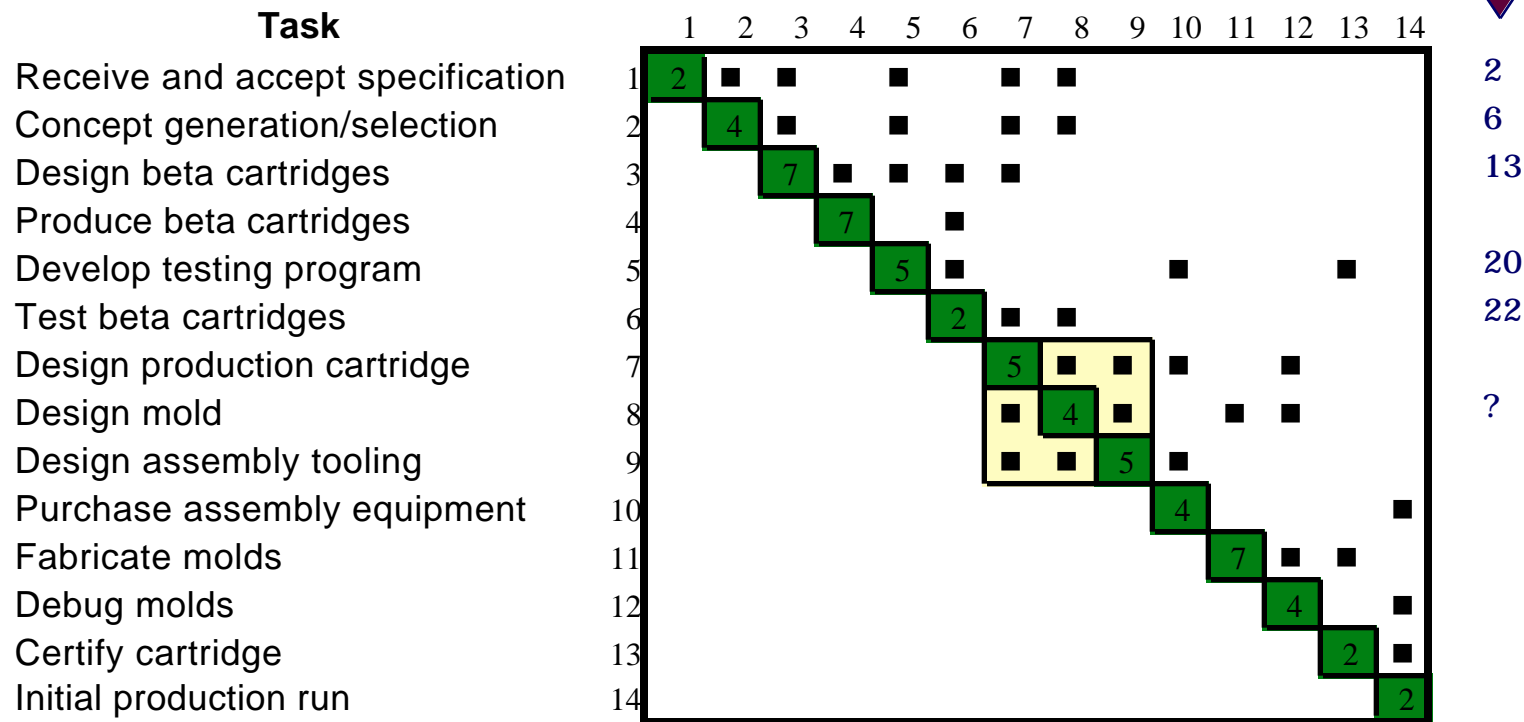


Diagram adapted from Ulrich and Eppinger, *Product Design and Development*, McGraw-Hill, 1995, p. 262

Kodak “Cheetah” Project

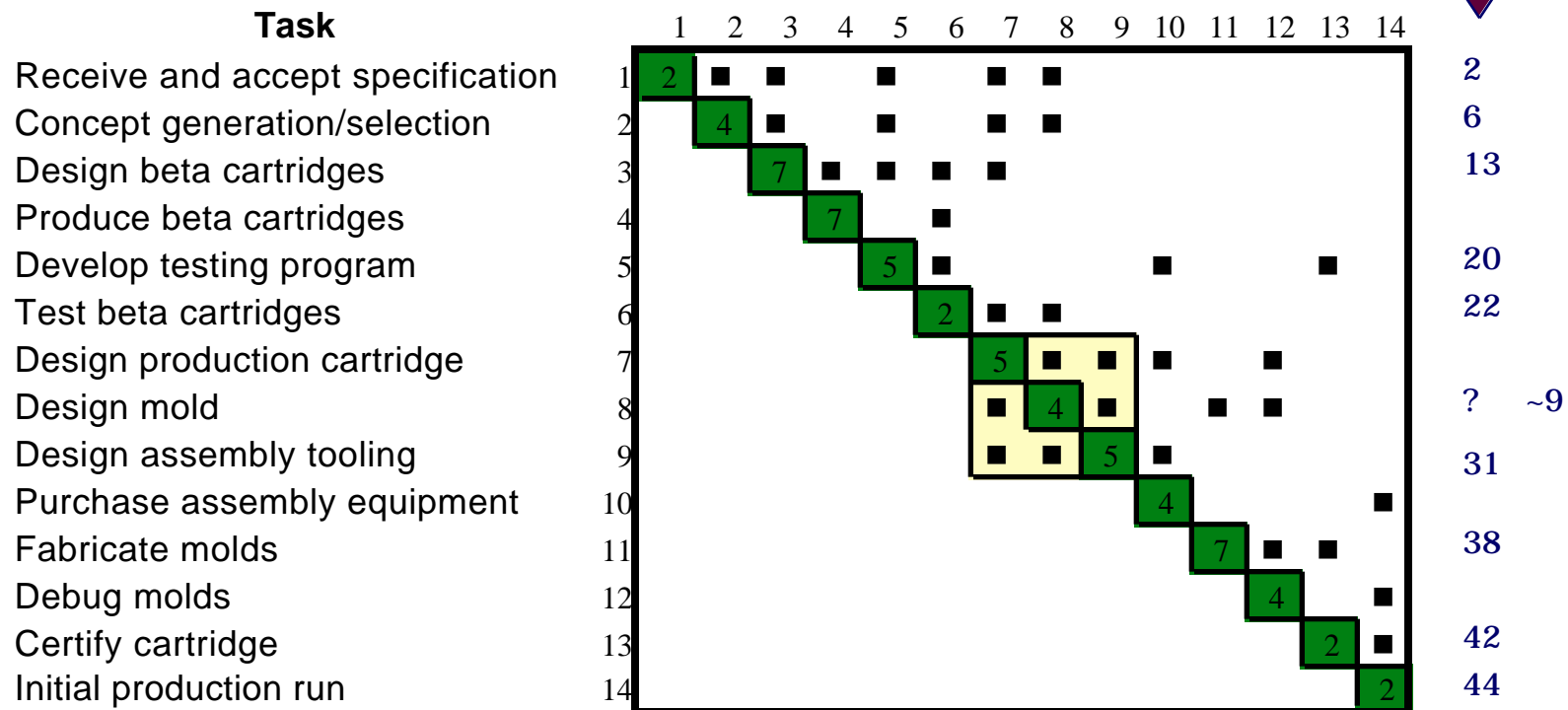


Diagram adapted from Ulrich and Eppinger, *Product Design and Development*, McGraw-Hill, 1995, p. 262

DSM Provides Method to Meet Cycle Time Reduction Challenges

- Shows where resources must flow
- Examine effects of changes
 - Unstable requirements
 - “Creeping elegance”
- Exposes long and varied subprocess pipelines
- Shows where to check for lack of task coordination
- Illuminates highly coupled activities
- Shows schedule impacts of delayed decisions
- Exhibits likely iterations and rework chains
- Helps ensure schedule not too ambitious

Applications and Next Steps

- **Applications**

- Process task/activity sequencing
- Cycle time and schedule risk reduction
- Process visualization, representation, and reengineering
- David Grose: applications at Boeing

- **Next Steps**

- Product Development Focus Group (tomorrow)
- Ongoing research to expand applications

- **Schedule Risk**
 - *Uncertainty in ability of program to develop a design that meets desired quality levels by certain points in time*
 - *... and the consequences thereof*
- **Performance Risk**
 - *Possibility that a new design will not meet desired quality criteria within cost and/or schedule expectations*
 - *... and the consequences thereof*
- **Technology Risk**
 - a subset of performance risk; the possibility that a technology will not provide anticipated performance benefits within cost and/or schedule expectations and the consequences thereof*
- **Development Cost Risk**
 - the possibility that designing to desired specifications within schedule expectations will cost more than anticipated and the consequences thereof*
- **Market Risk**
 - the possibility that the chosen “design to” specifications will not be of anticipated utility to the market and the consequences thereof*

Representative Schedule Risk Drivers

- **Overly optimistic schedule**
- **Lack of personnel or other resources to complete activities**
- **Multiple refinements of analyses or repetition of tasks required to improve design performance relative to specifications**
- **Schedule rigidly dictated/constrained by available budget profile**
- **Subtask or subprocess duration is too long**
- **Subtask or subprocess duration is unpredictable**
- **Highly coupled activities must coordinate repeated, bilateral information exchange**
- **Decisions take too long to get made**
- **Information required to begin some tasks is not available when task begins**
- **Requirements change forces rework**
- **Inadequate development cost budget**
- **Design mistakes cause rework**
- **Long lead time tasks force early decisions**
- **Task content or ramifications are inadequately understood**
- **Organization is inflexible regarding adaptation to schedule modifications**
- **Personnel and/or leadership turnover causing delays**
- **Task set or statement of work is underestimated**