ESD.36 System Project Management

Lecture 7

The Rework Cycle

Instructor(s)

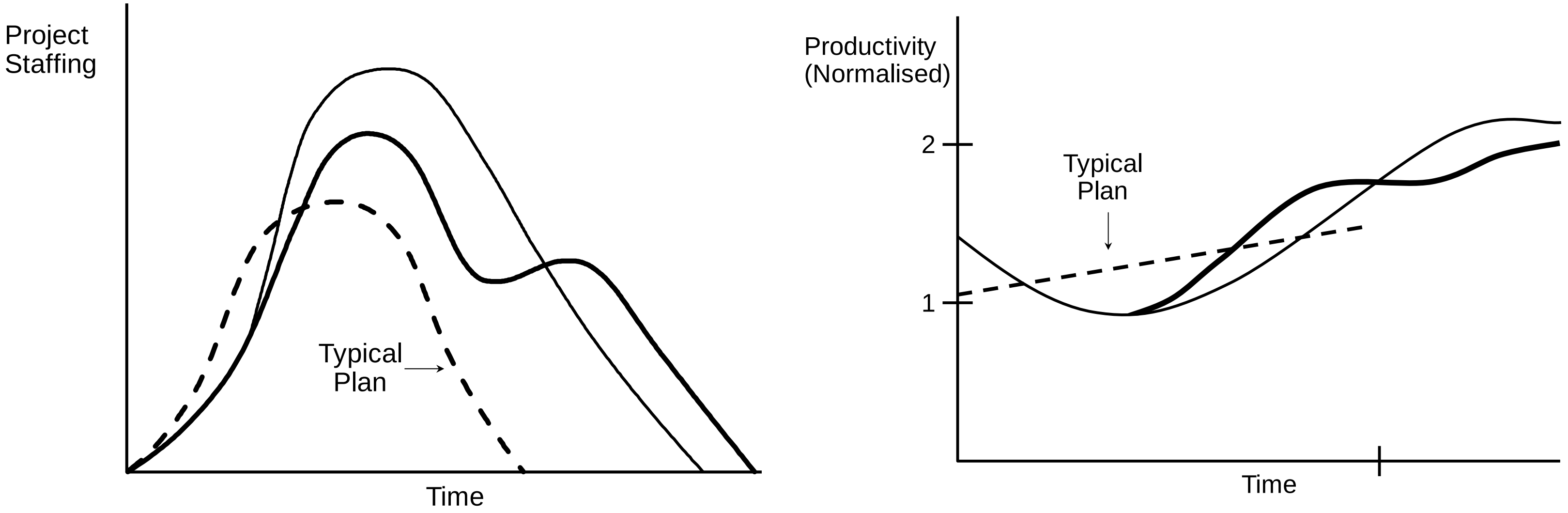
Dr. James Lyneis

Massachusetts Institute of Technology

Today's Agenda

➡
- Overview: Causes of Project Dynamics
- The Rework Cycle
- Integrating Tools in Project Planning
- Simple Model of Project Dynamics, Pt. 1

Typical Behavior Modes on a Project



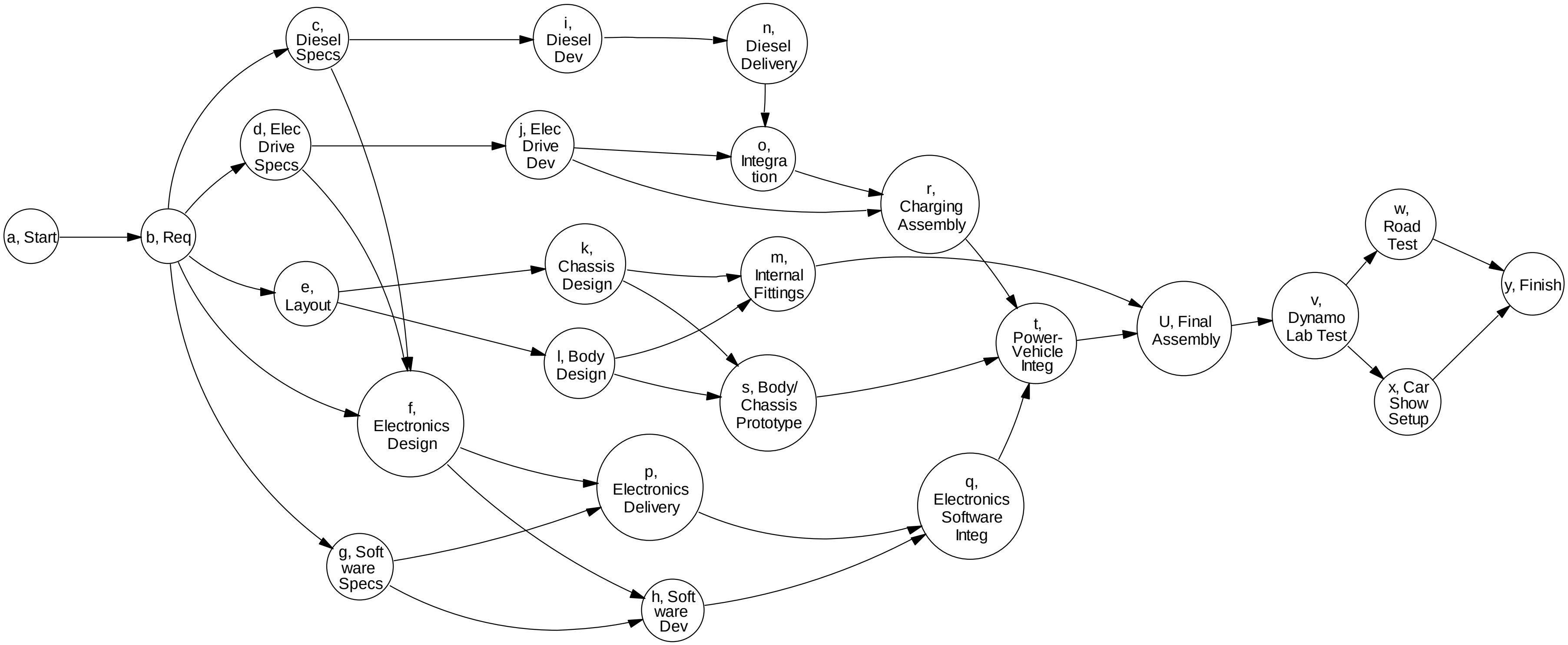
Drivers of Project Dynamics

- The "rework cycle"
- Feedback effects
  - Negative, controlling
  - Positive, re-enforcing, often "vicious circles"
- Knock-on or domino effects within or between work phases
- *Knock-on or domino effects between projects*

Today’s Agenda

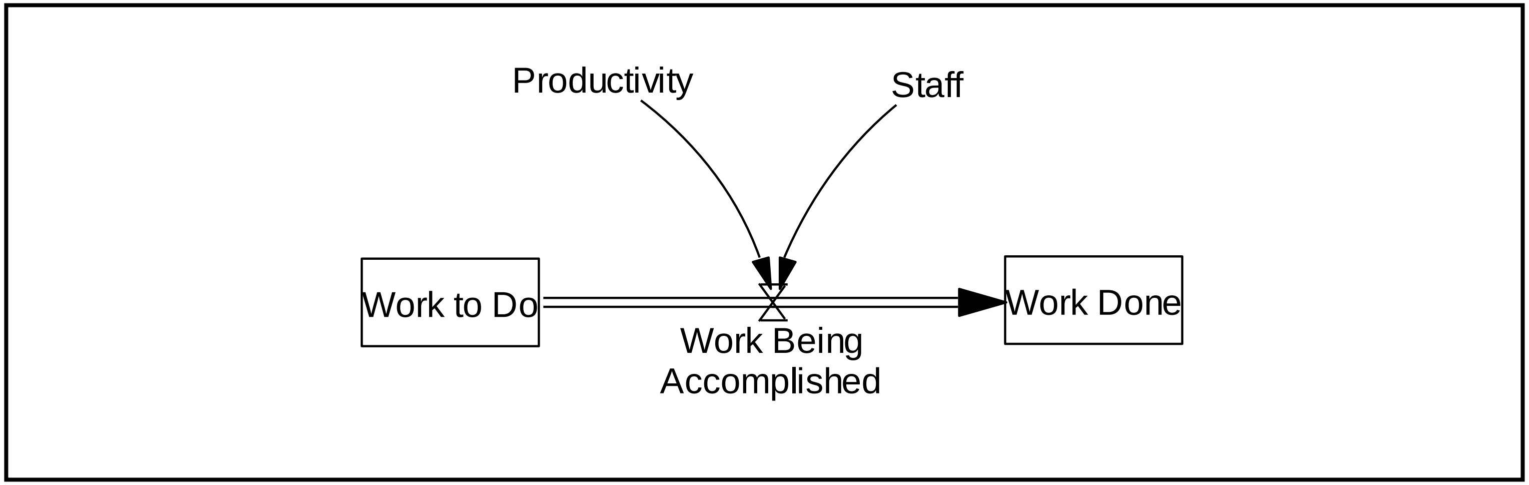
- Overview: Causes of Project Dynamics
- ➡ The Rework Cycle
- Integrating Tools in Project Planning
- Simple Model of Project Dynamics, Pt. 1

Network Diagram for NMM Case (from 2010 Homework #1)

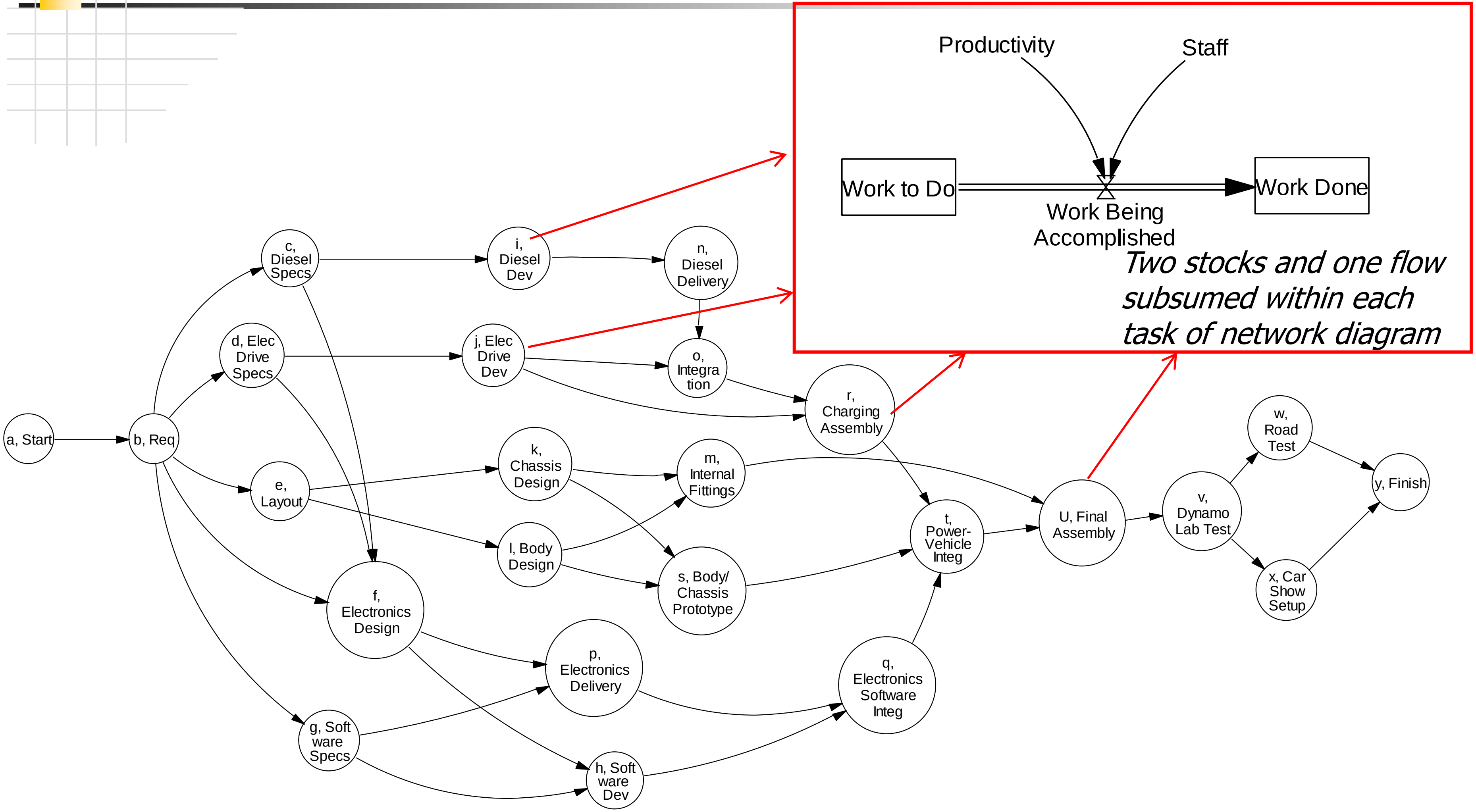


The Network View of a Program – Task Accomplishment

Task Accomplishment – Task X

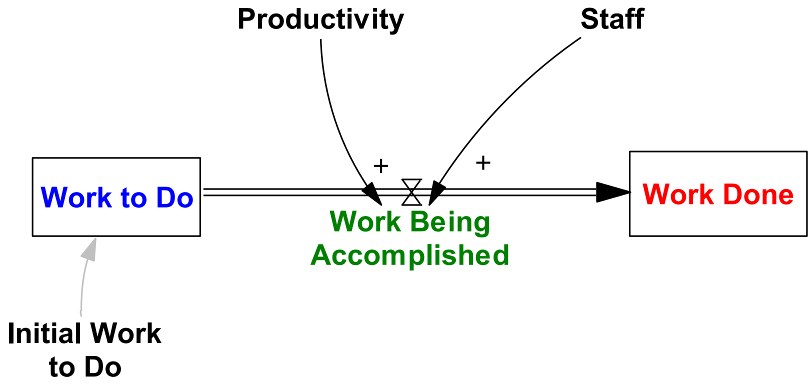


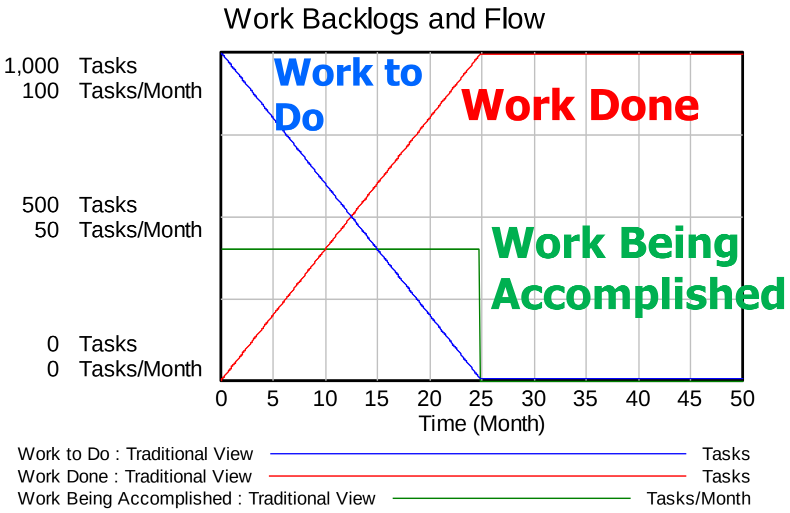
Network Diagram for NMM Case



Two stocks and one flow subsumed within each task of network diagram

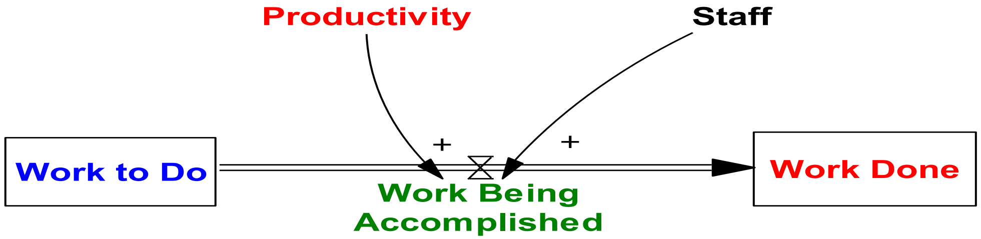
Aggregate Model of Network View

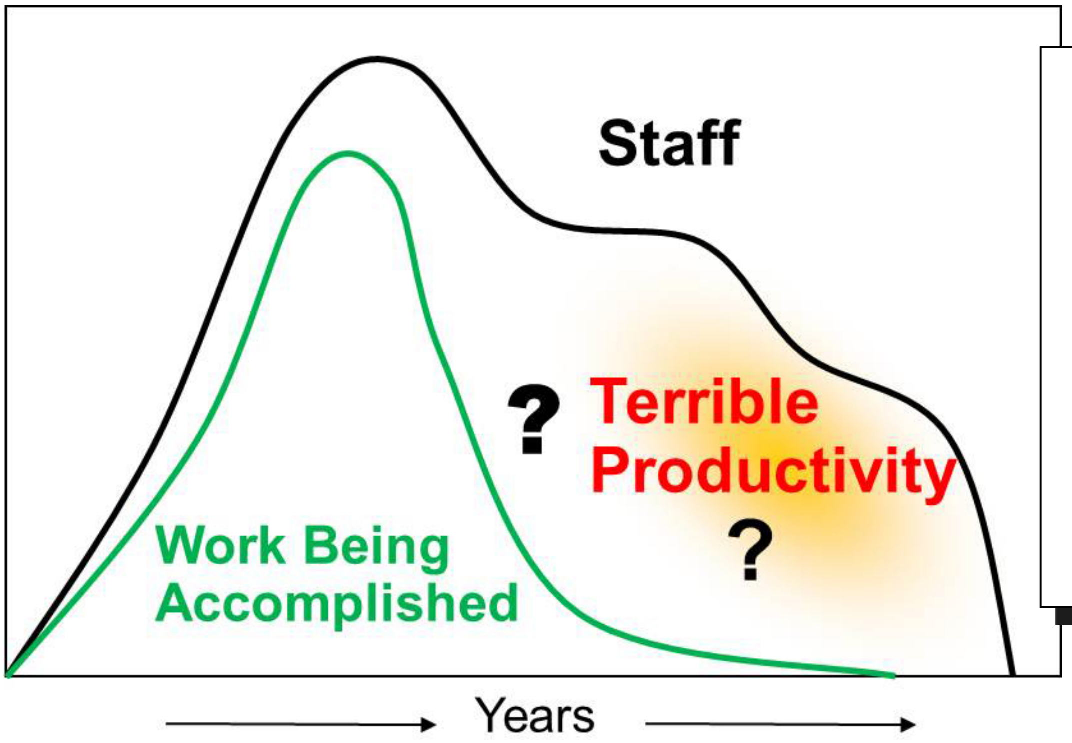


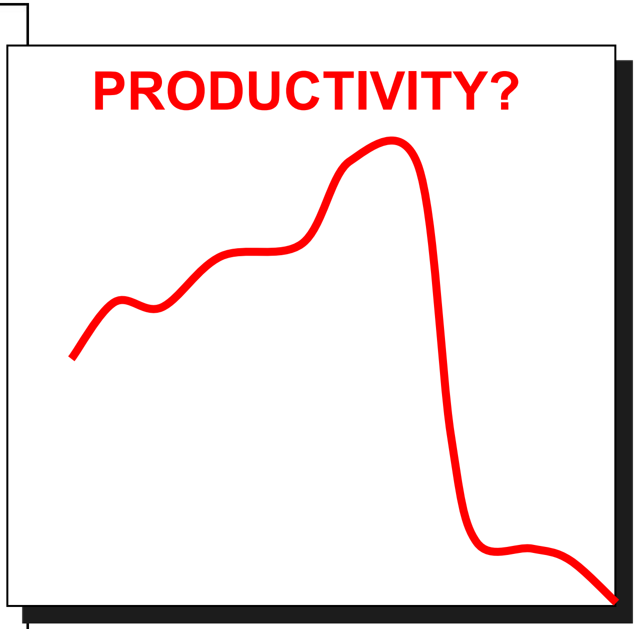


Initial Work to Do = 1000 tasks
Productivity = 1 task/mo/person
Staff = 40 people

Data Often Tell a More Complex Story

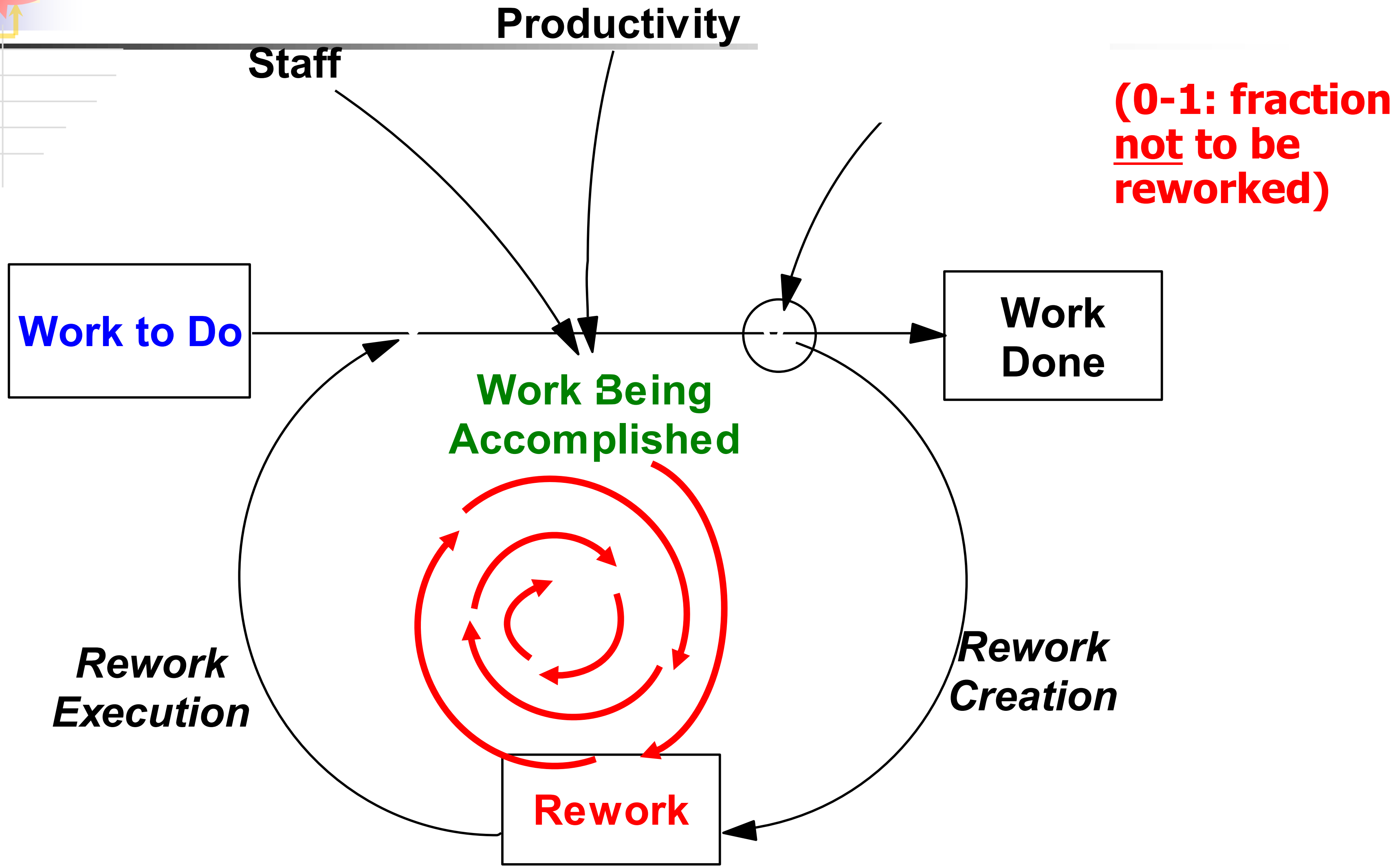






OR...

Effort is being spent elsewhere …



Definitions

- Productivity --

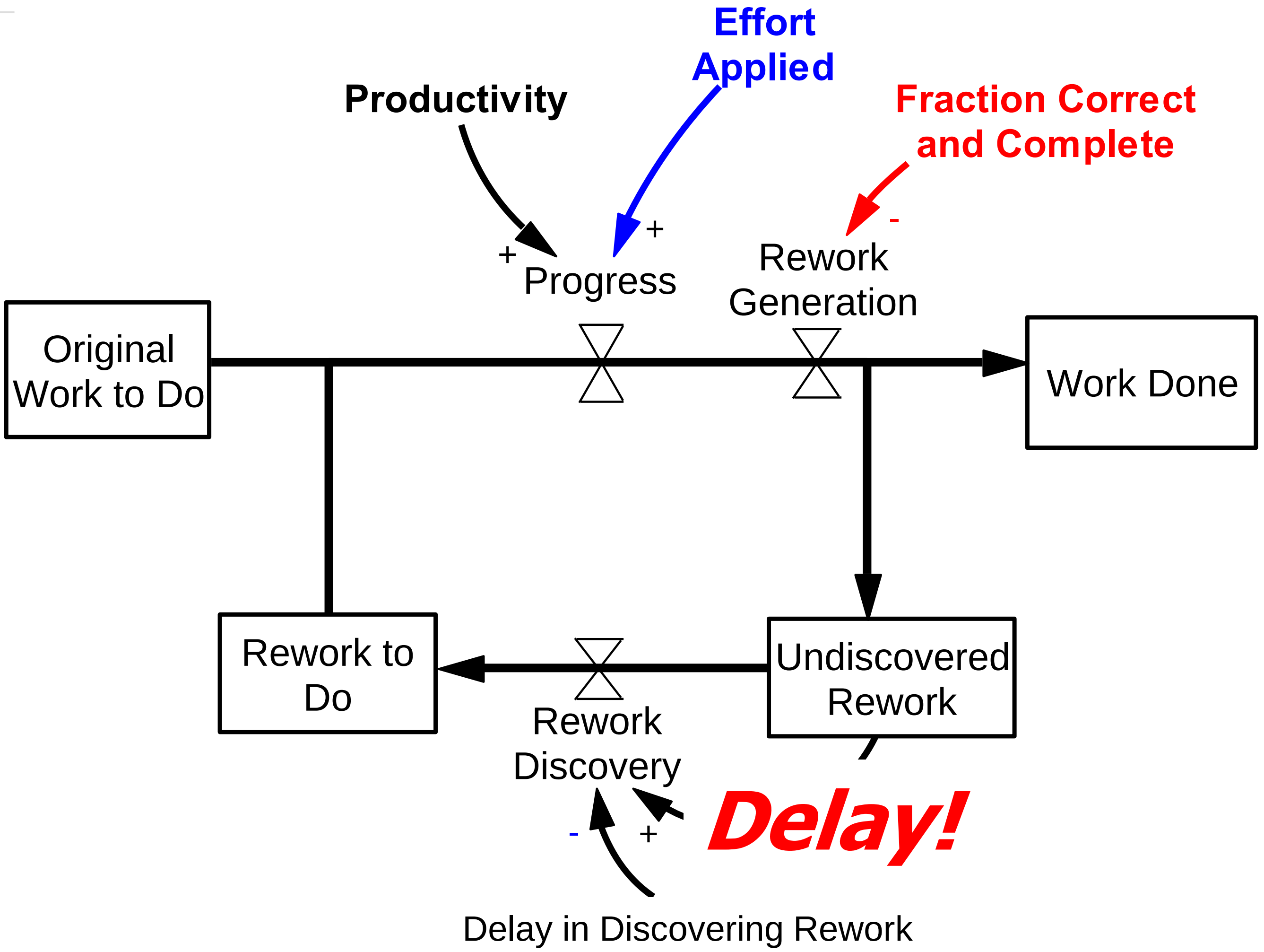
  Work accomplished per hour of effort, regardless of completeness or correctness

- Fraction Correct & Complete --

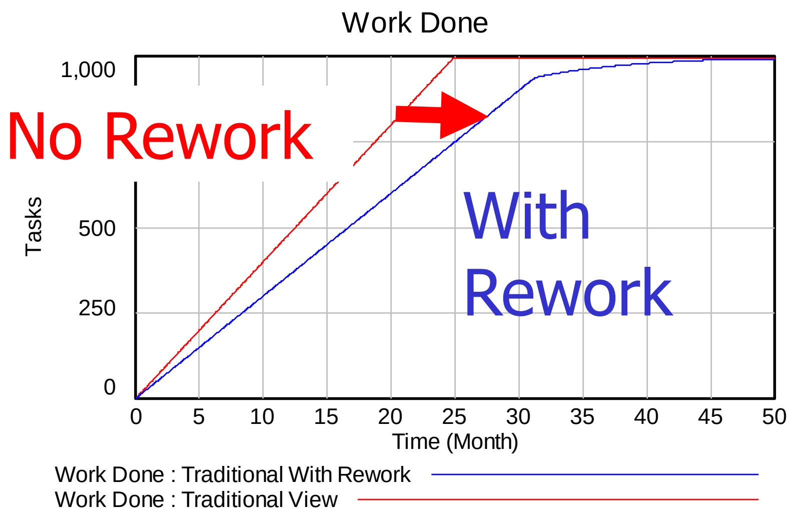
  Fraction of work just accomplished that will not need rework. “Work Quality”

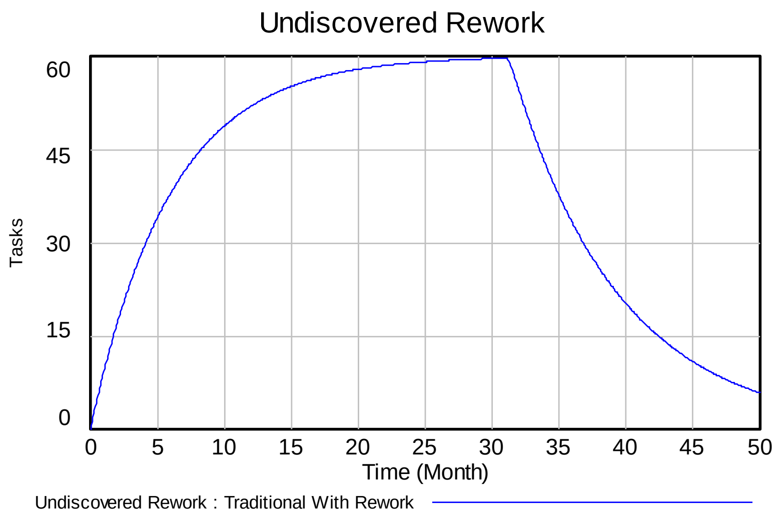
  Later we will revisit the iron triangle to consider “delivered quality.”

The Rework Cycle

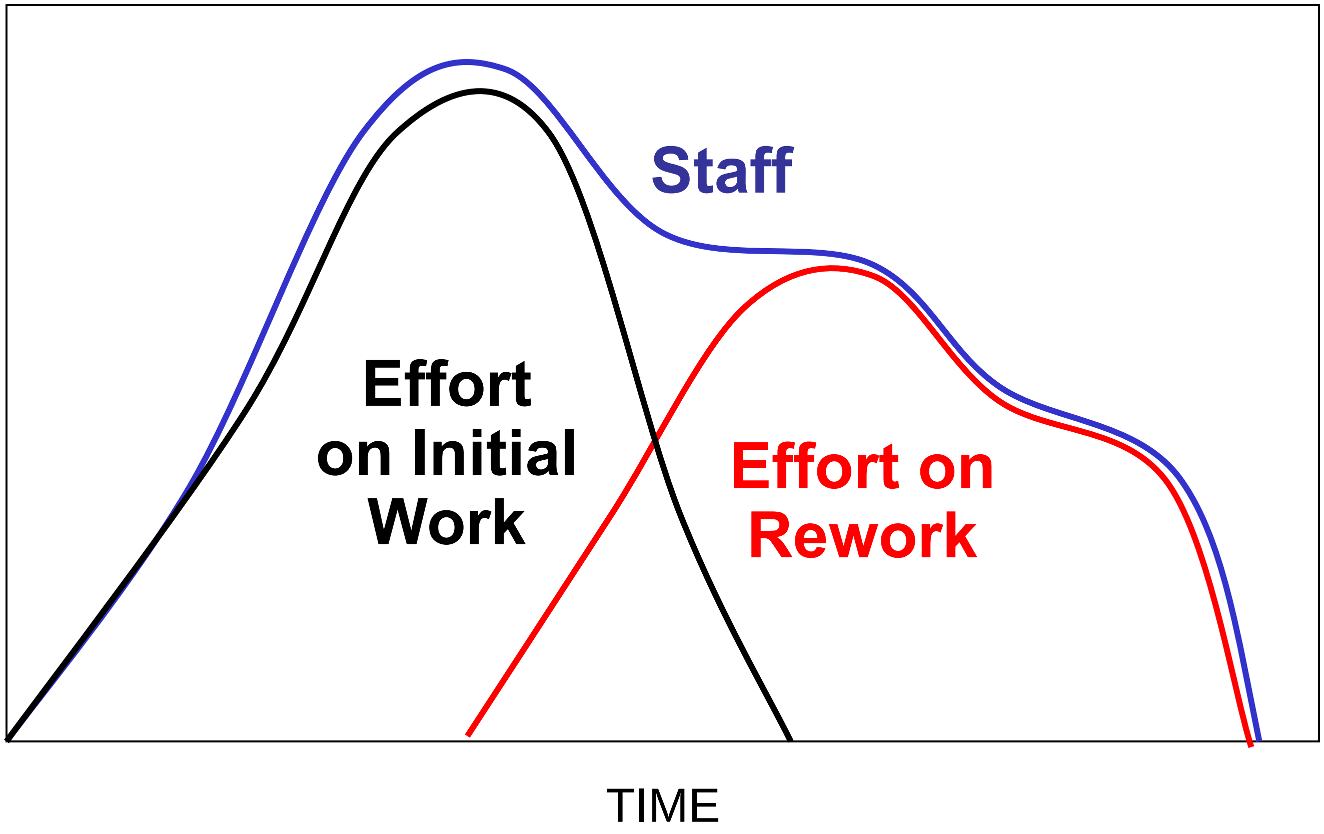


Adding Rework Delays Completion …

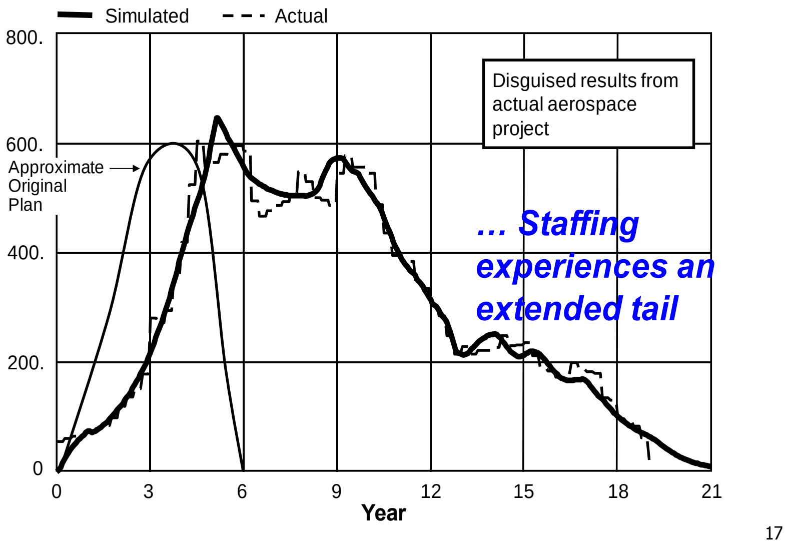




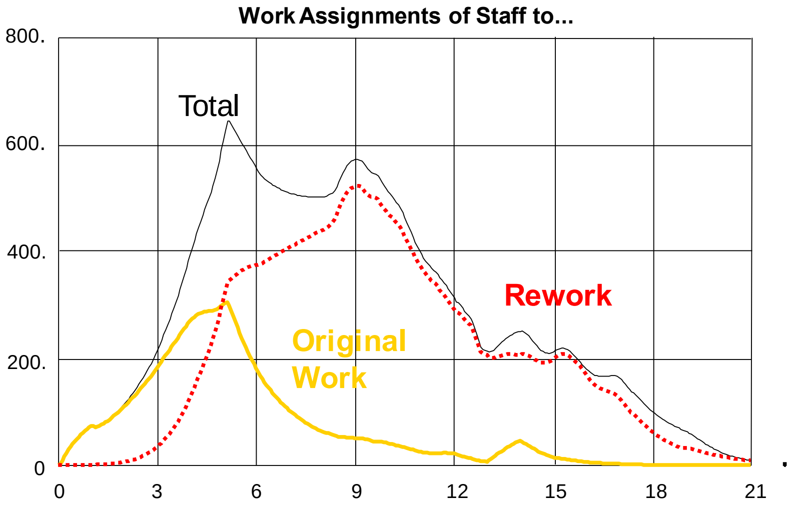
So, What Happens on Projects?



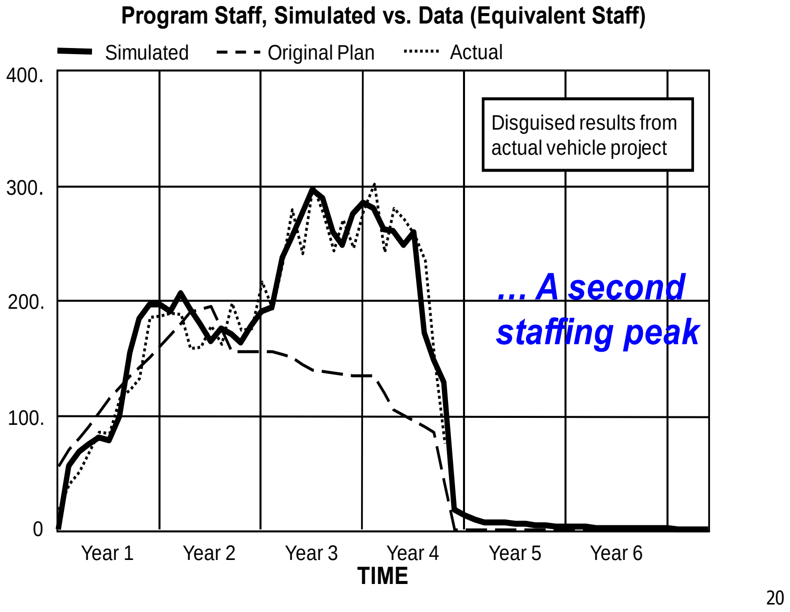
Rework creates and extended staffing tail

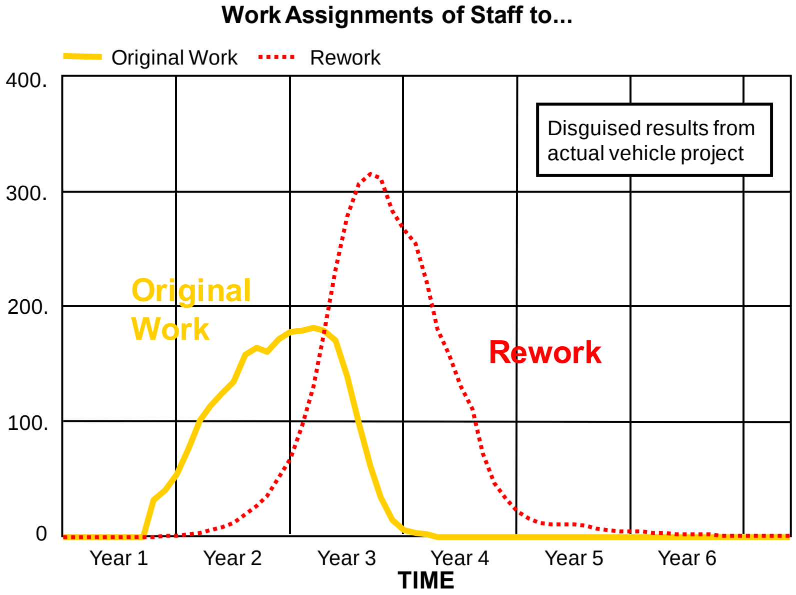


… Staffing experiences an extended tail

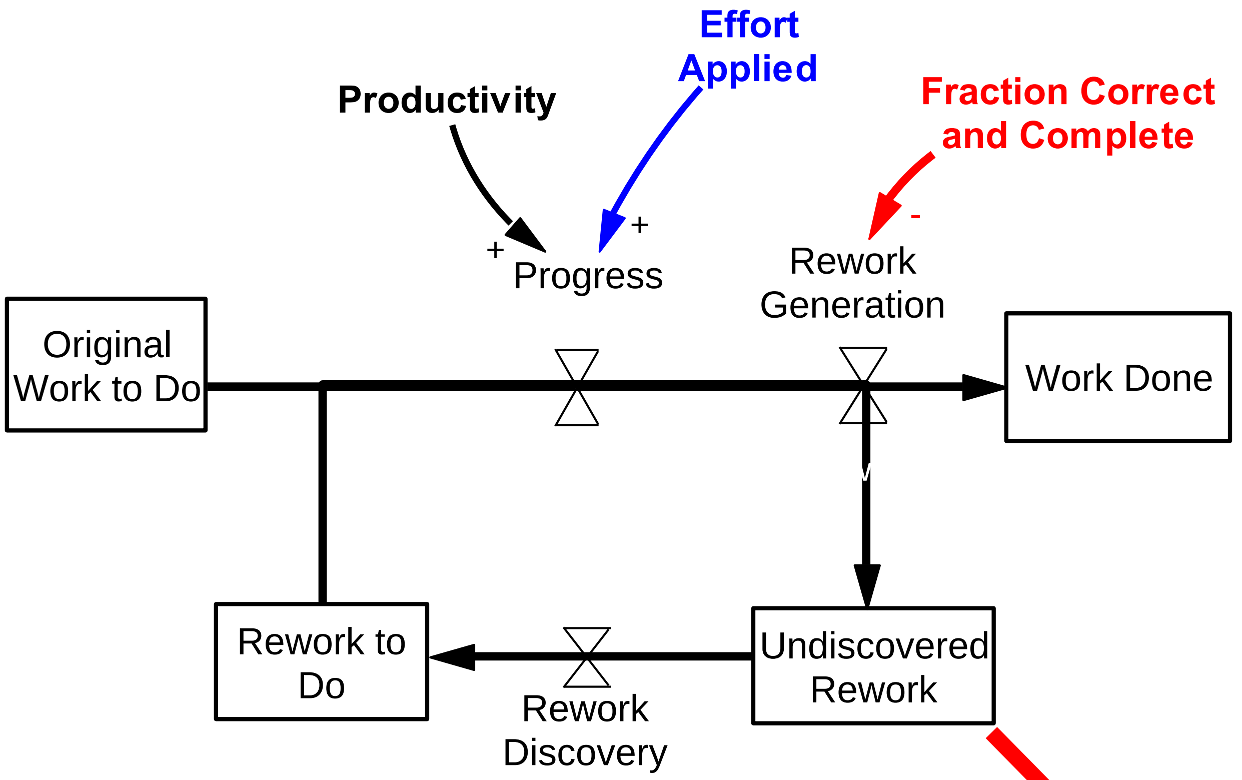


… or a second staffing peak





Consequences of Undiscovered Rework

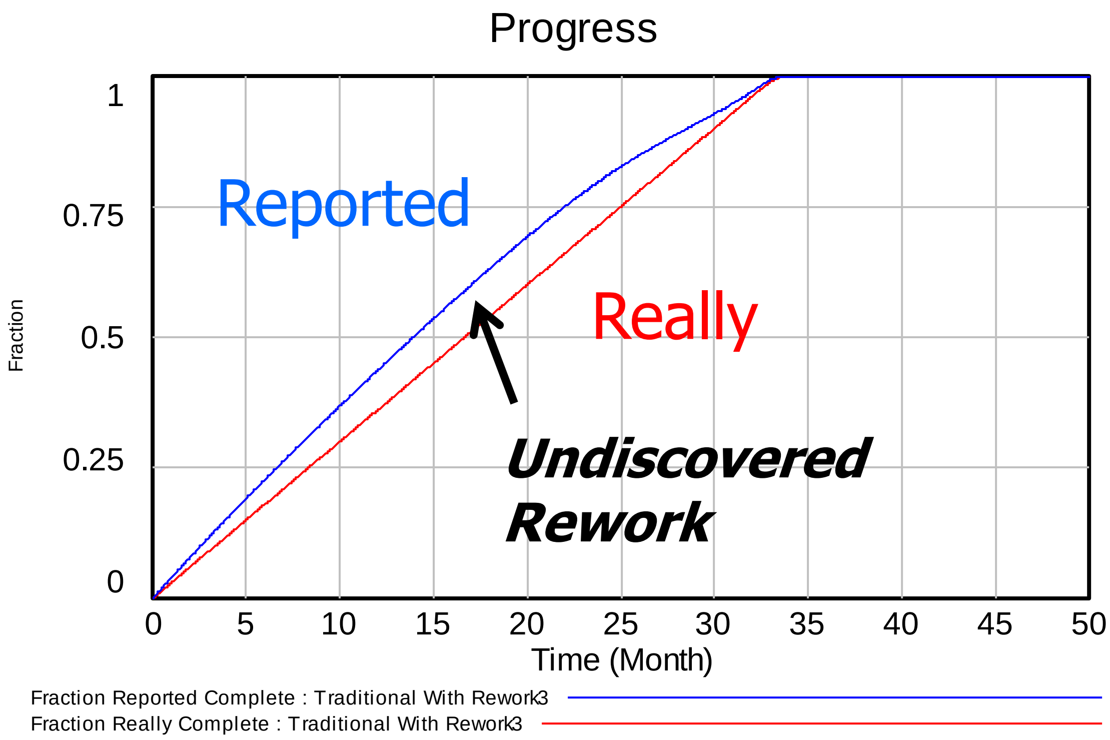


1.Overestimates of Progress

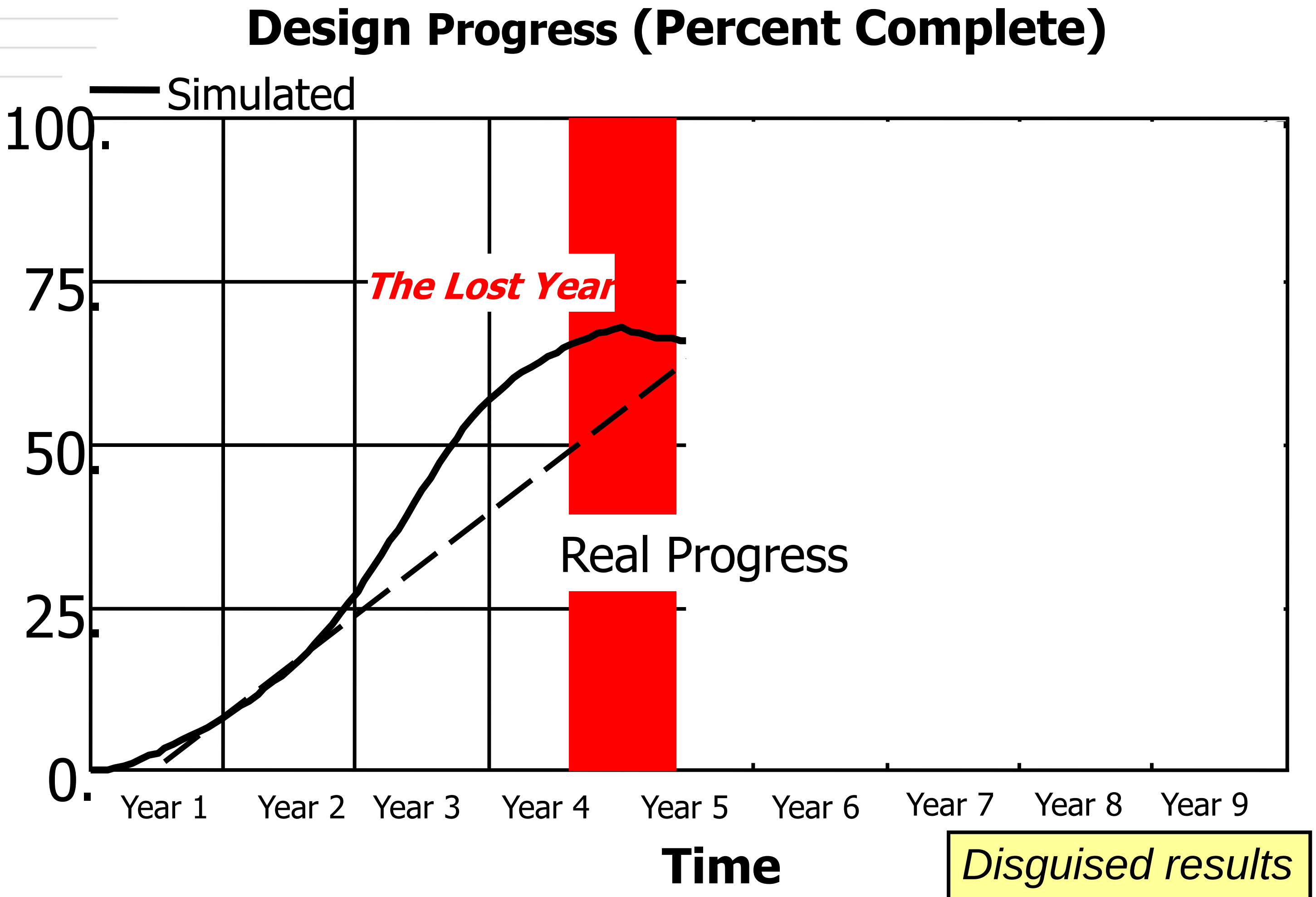
Additional Metrics (Used in Model)

- Fraction Reported to be Complete = (Work Done + Undiscovered Rework) / Initial Work to Do
- Fraction of Work Really Complete = (Work Done) / Initial Work to Do

Undiscovered rework generates progress-measuring errors …



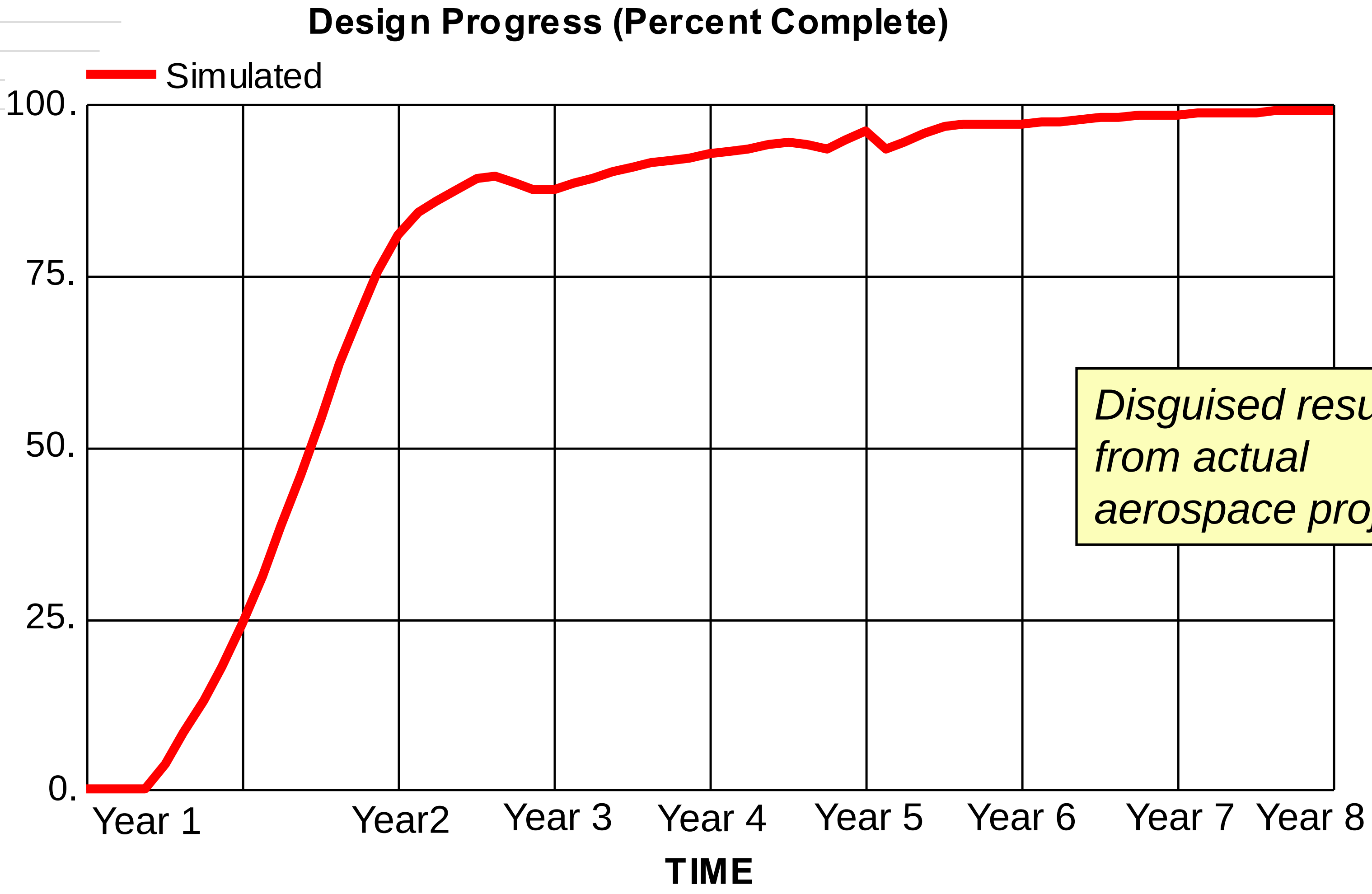
Rework Cycle Creates "The Lost-Year"...



Disguised results from actual aerospace project

Source: Ken Cooper

and the "90% Syndrome" ---



Disguised results from actual aerospace project

Survey Question 1

Does your organization ***measure*** rework?

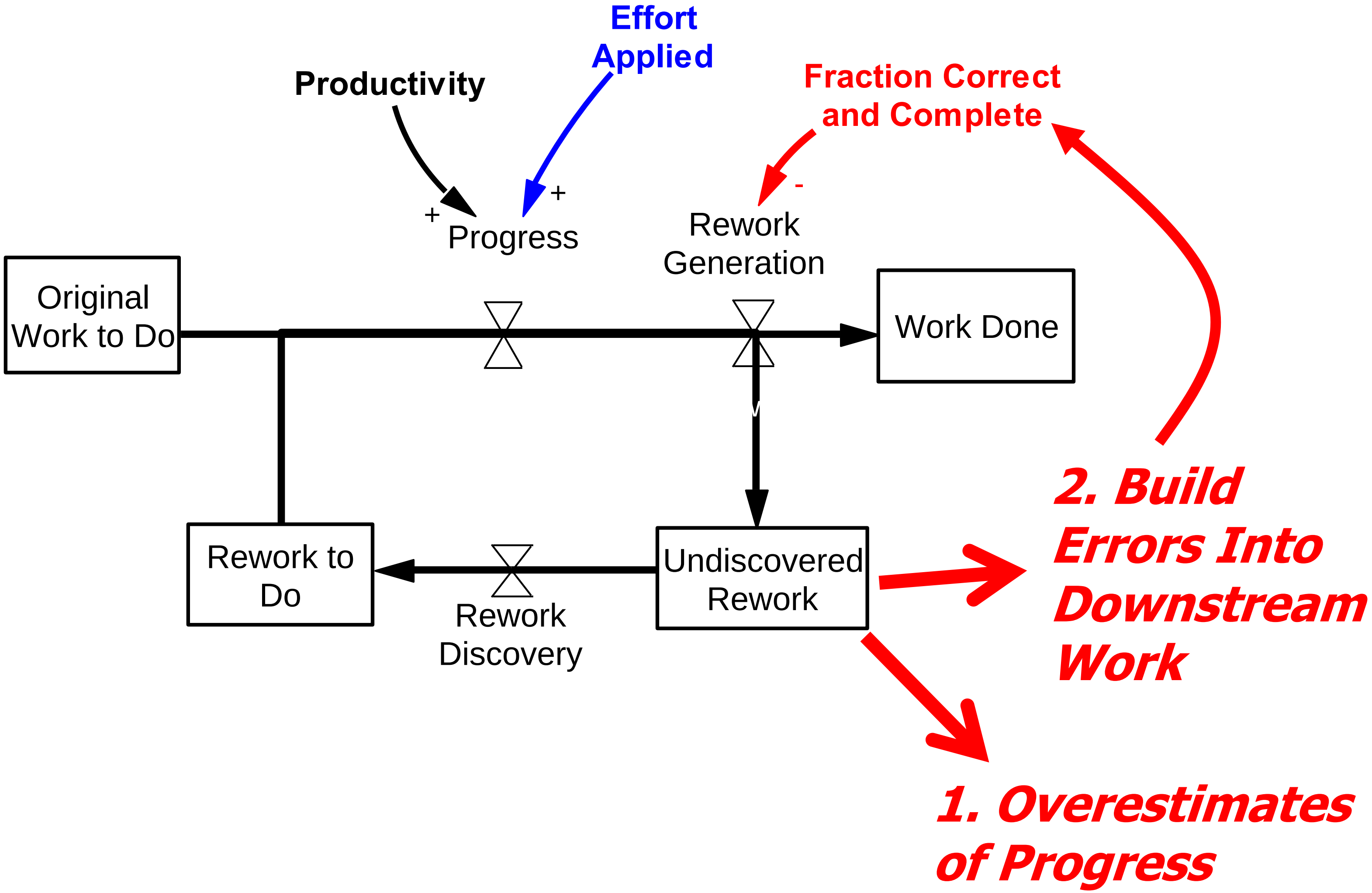
1. Yes, we keep data on how much rework is being ***discovered***
2. Yes, we keep data on how much ***work being done*** is rework (vs original work)
3. Yes, both 1 and 2
4. No, we do not keep data on rework being discovered

Survey Question 2

Does your organization recognize rework in executing the project?

1. Yes, by adjusting progress estimates to reflect expected remaining undiscovered rework
2. Yes, by building rework tasks into the project graph
3. Both 1 and 2
4. No, we only react as rework is discovered

Consequences of Undiscovered Rework



2. Build Errors Into Downstream Work

1. Overestimates of Progress

Survey Question 3

How significant is the "Errors on Errors" feedback on typical projects in your organization?

1. Very
2. Modest
3. Weak
4. None

Units of Work – SD Model

- “Tasks” -- ideally, of uniform size and fungible
- In actual applications, might use things like:
  - Drawings, work packages, lines of code, …
  - Tons of concrete, feet of steel, feet of wiring, ...
  - Parts designed
  - Sometimes %
- Can represent “precedence” constraints

"Quality"

Need to distinguish between:

- Instantaneous "work quality" (fraction correct and complete)
- Delivered "quality" – remaining "bugs" (undiscovered rework) when product shipped
- Product capabilities, "fit and finish" (a part of scope in model)

Fraction Correct and Complete

- Represents unplanned iterations; planned iterations are separate tasks
- Sources of errors many fold:
  - Mistakes from inexperience, fatigue, …
  - Technical complexity/uncertainty
  - Work done “correctly” but ultimately needing rework because it builds on
    - Incorrect prior work
    - Assumptions about technology or customer requirements which prove incorrect

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.

Separate tasks in the rework cycle

Unplanned Iteration

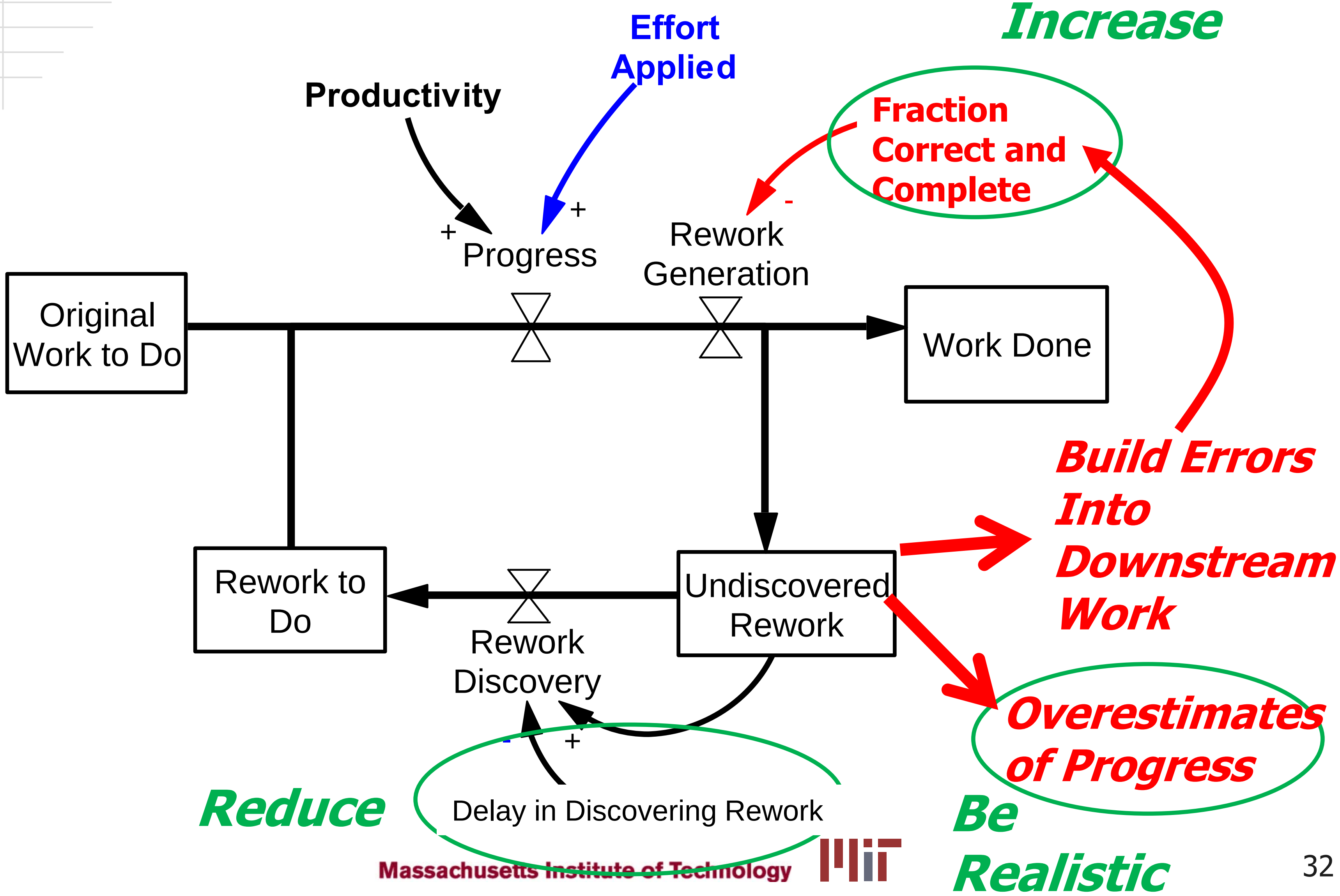
- Caused by errors and/or unforeseen problems.
- We generally cannot predict which unplanned iterations will occur.
- Unplanned iterations should be minimized using risk management methods.

"Quality" problems in the rework cycle

Qualitative Insights

- Undiscovered Rework is one of the most important single factors driving schedule and budget overruns
- Most management reporting systems overestimate real progress and discourage reporting of rework

Lessons: Recognize the rework cycle and minimize its consequences



Dynamics of Project Performance

The "rework cycle"

- Fraction correct and complete ✓
- Undiscovered rework ✓

Feedback effects on productivity and fraction correct (Next class)

- Negative, controlling
- Positive, re-enforcing, often "vicious circles"

Knock-on effects between work phases (Next)

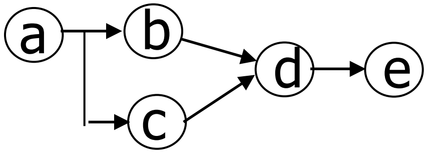
- Availability and quality of work products
- Progress to discover upstream rework

Today's Agenda

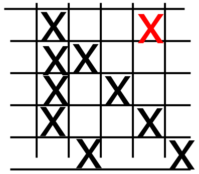
- Causes of Project Dynamics
- The Rework Cycle
- ➡ Integrating Tools in Project Planning
- Simple Model of Project Dynamics, Pt. 1

Fundamental Approaches

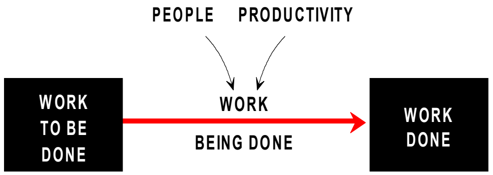
- Network-based (graph theory) methods
  - CPM, PERT, ….
  - Task is a node or an arc



- Matrix-based methods
  - DSM - Tasks are columns and rows
  - Interrelationships are off-diagonal entries

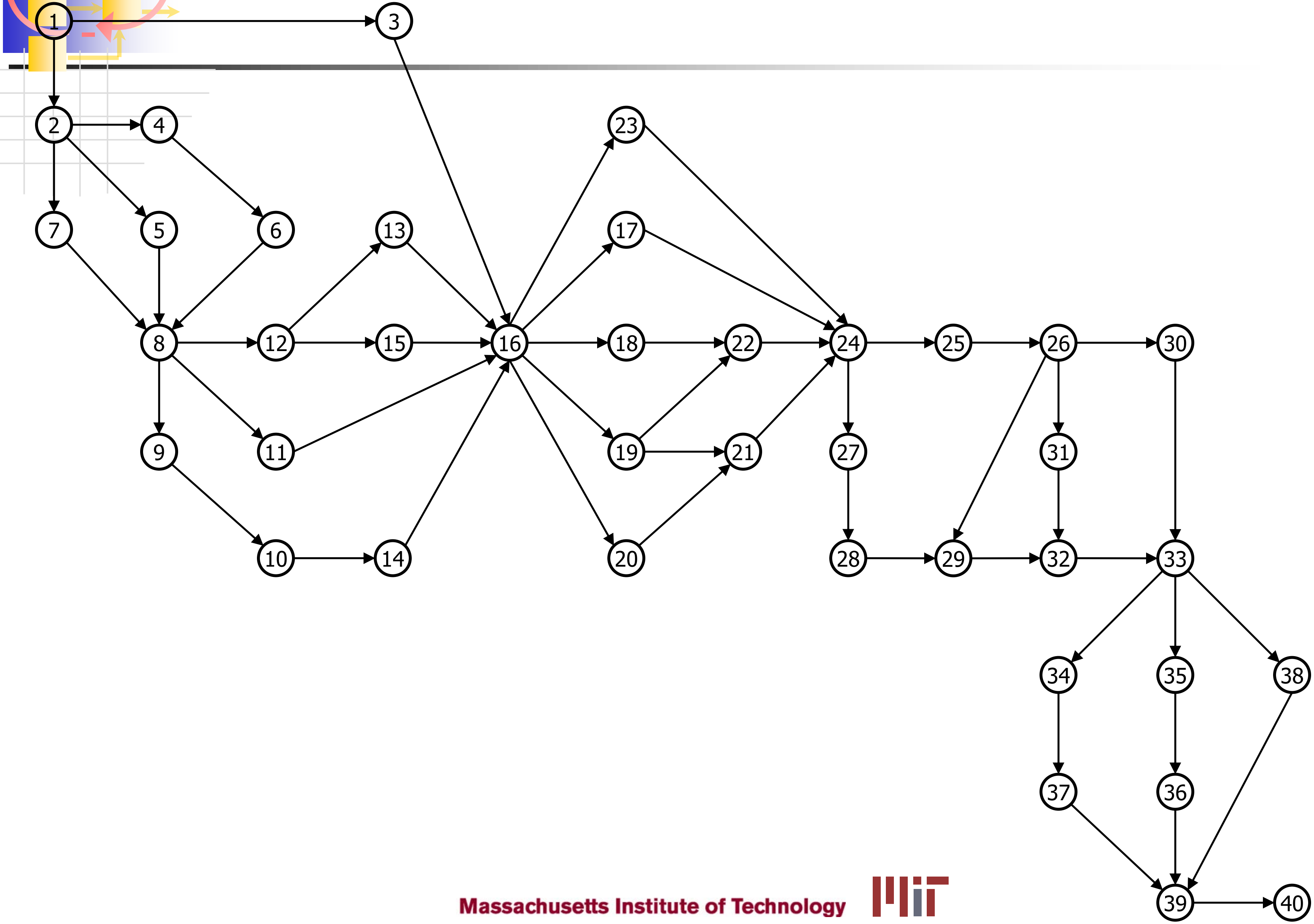


- System Dynamics
  - Feedback loops, causal relationships
  - Stocks and flows simulation
  - Tasks that are done or waiting to be done are stocks – "amount of work"
  - Doing project work causes a "flow"



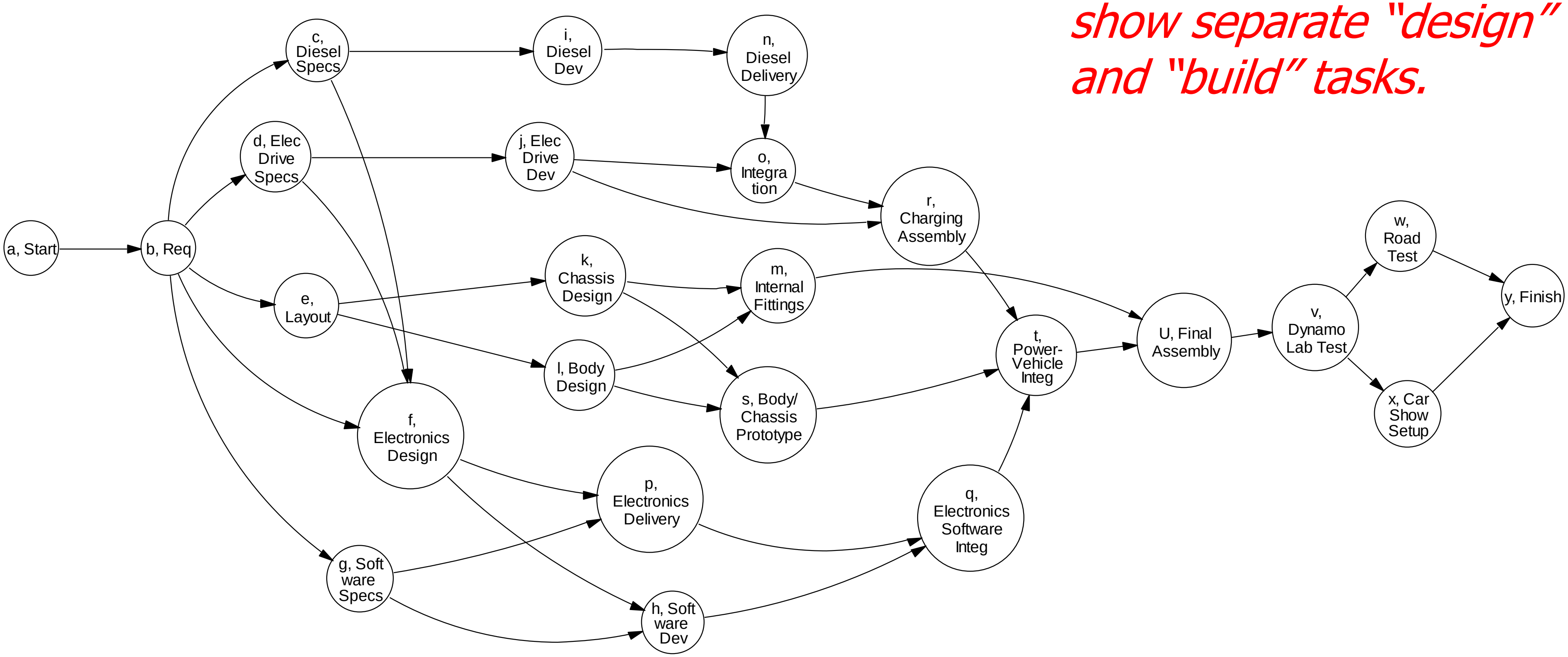
What have we learned thus far?

Network View of *CityCar* Project

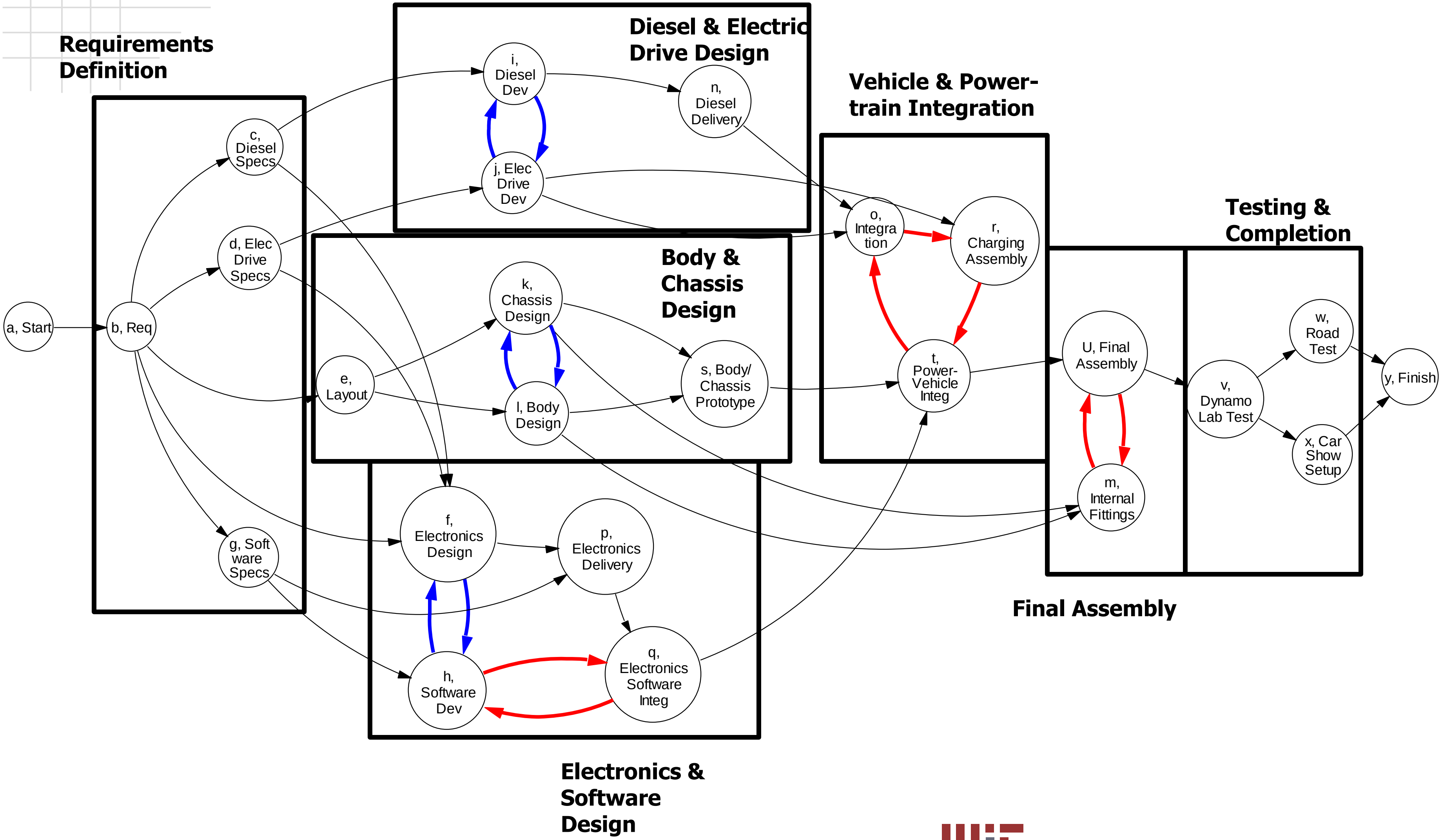


Network Diagram for NMM Case (from 2010 Homework #1)

Note: Diesel, Elec Drive, and Software do not show separate "design" and "build" tasks.

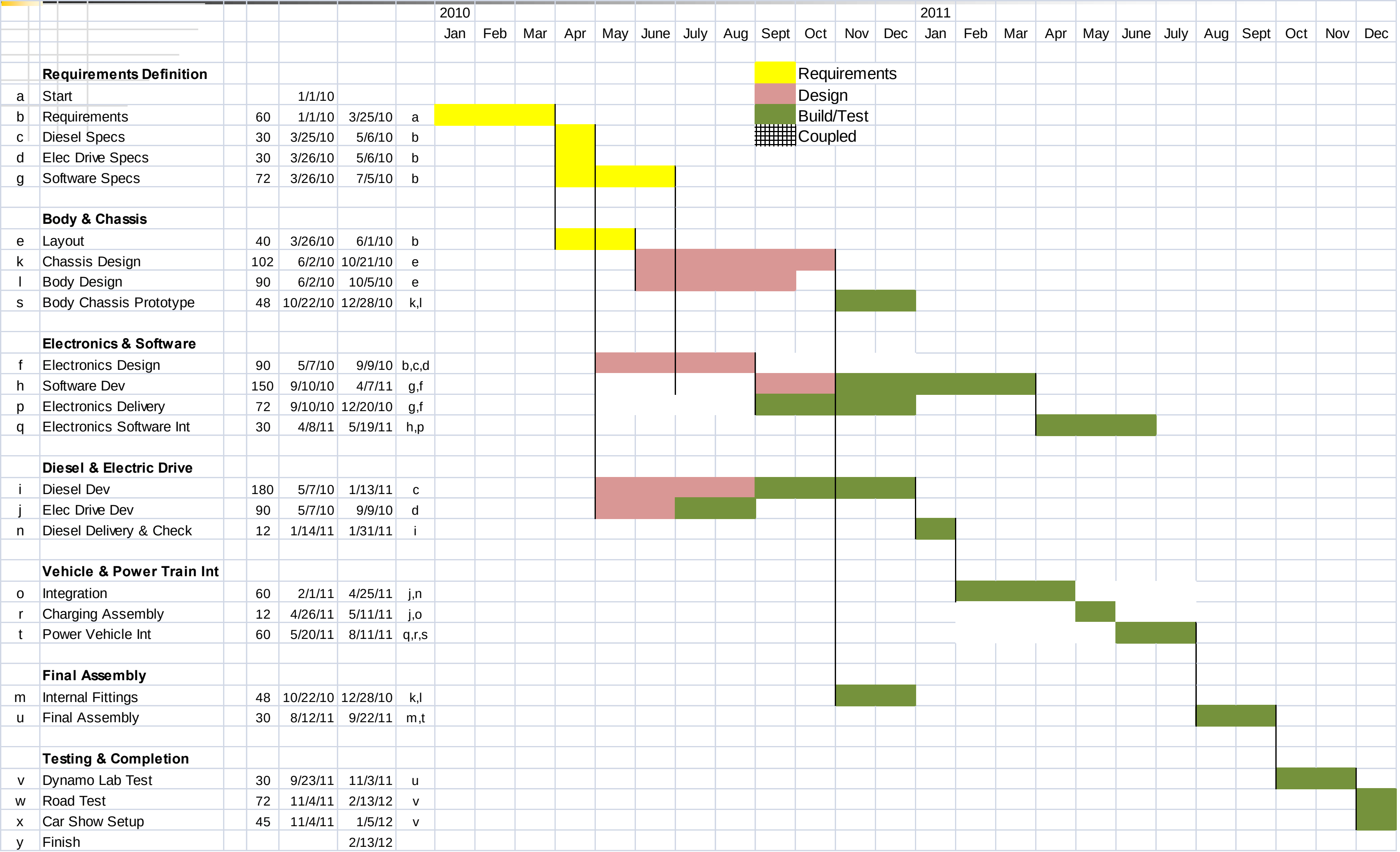


Organizing the Tasks by “Metatask”



“Gantt” Chart with Metatasks Shown

Note: dates approximate; JML estimates of Diesel & Electric, Software Design/Build Split

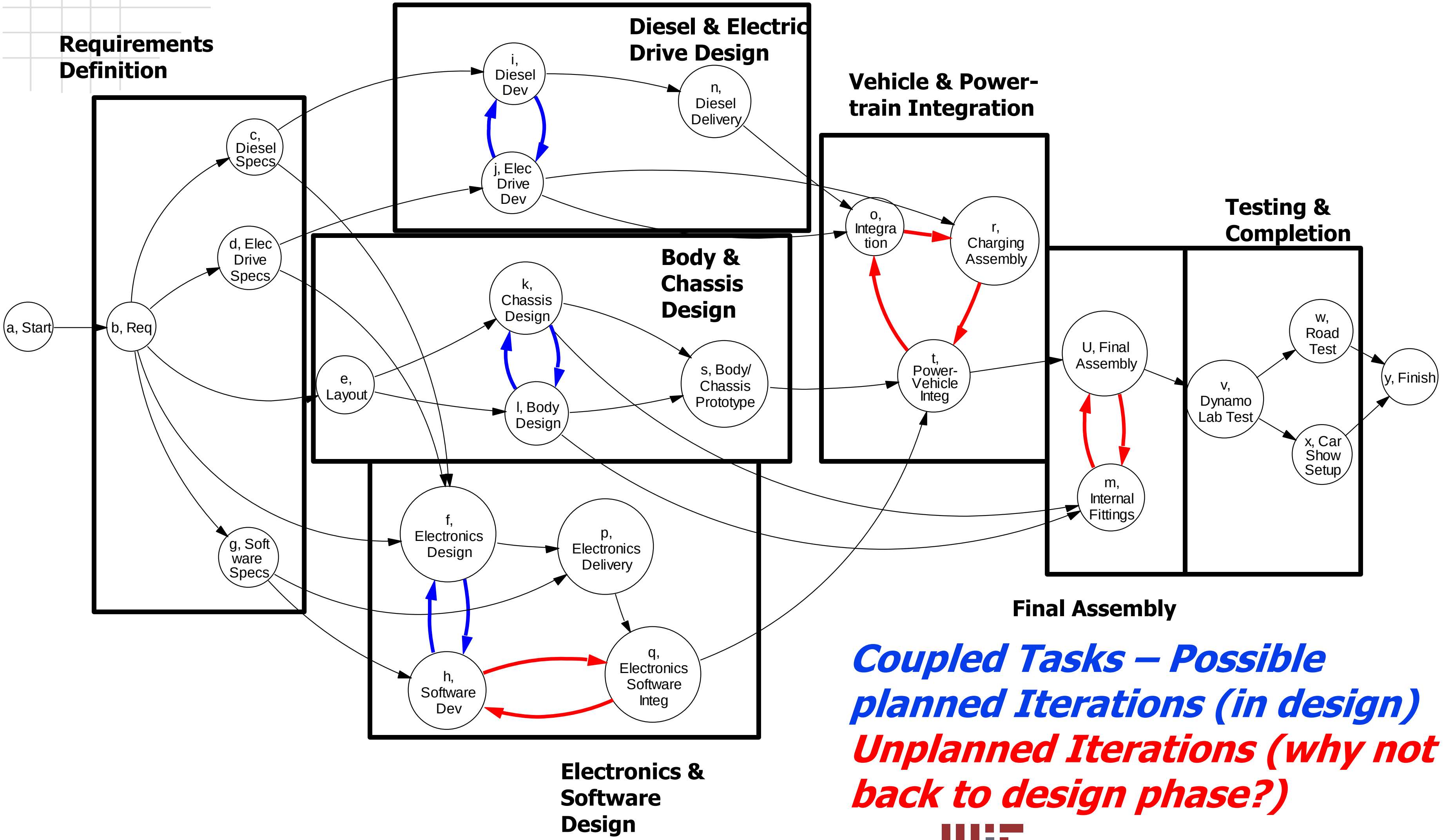


	Requirements Definition				
a	Start		1/1/10		
b	Requirements	60	1/1/10	3/25/10	a
c	Diesel Specs	30	3/25/10	5/6/10	b
d	Elec Drive Specs	30	3/26/10	5/6/10	b
g	Software Specs	72	3/26/10	7/5/10	b
	Body & Chassis				
e	Layout	40	3/26/10	6/1/10	b
k	Chassis Design	102	6/2/10	10/21/10	e
l	Body Design	90	6/2/10	10/5/10	e
s	Body Chassis Prototype	48	10/22/10	12/28/10	k,l
	Electronics & Software				
f	Electronics Design	90	5/7/10	9/9/10	b,c,d
h	Software Dev	150	9/10/10	4/7/11	g,f
p	Electronics Delivery	72	9/10/10	12/20/10	g,f
q	Electronics Software Int	30	4/8/11	5/19/11	h,p
	Diesel & Electric Drive				
i	Diesel Dev	180	5/7/10	1/13/11	c
j	Elec Drive Dev	90	5/7/10	9/9/10	d
n	Diesel Delivery & Check	12	1/14/11	1/31/11	i
	Vehicle & Power Train Int				
o	Integration	60	2/1/11	4/25/11	j,n
r	Charging Assembly	12	4/26/11	5/11/11	j,o
t	Power Vehicle Int	60	5/20/11	8/11/11	q,r,s
	Final Assembly				
m	Internal Fittings	48	10/22/10	12/28/10	k,l
u	Final Assembly	30	8/12/11	9/22/11	m,t
	Testing & Completion				
v	Dynamo Lab Test	30	9/23/11	11/3/11	u
w	Road Test	72	11/4/11	2/13/12	v
x	Car Show Setup	45	11/4/11	1/5/12	v
y	Finish			2/13/12	

What Messages Come From DSM and SD regarding the network/Gantt plan?

- DSM ?
- SD ?
- How/when is rework discovered?

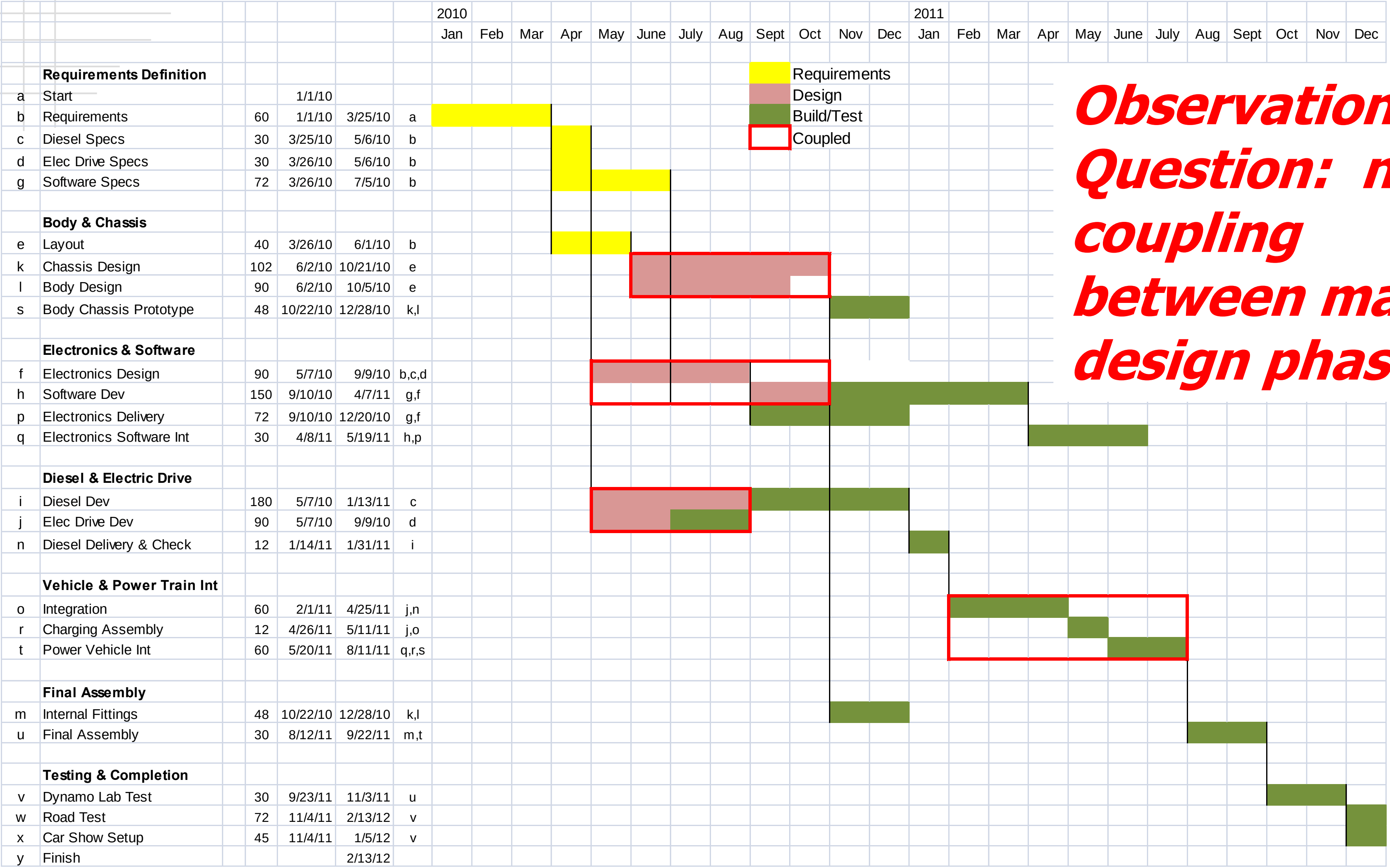
DSM Identifies Coupled Tasks & Areas of Unplanned Iterations (from Homework #2)



Coupled Tasks – Possible planned Iterations (in design)

Unplanned Iterations (why not back to design phase?)

"Gantt" Chart With Coupled Tasks Shown

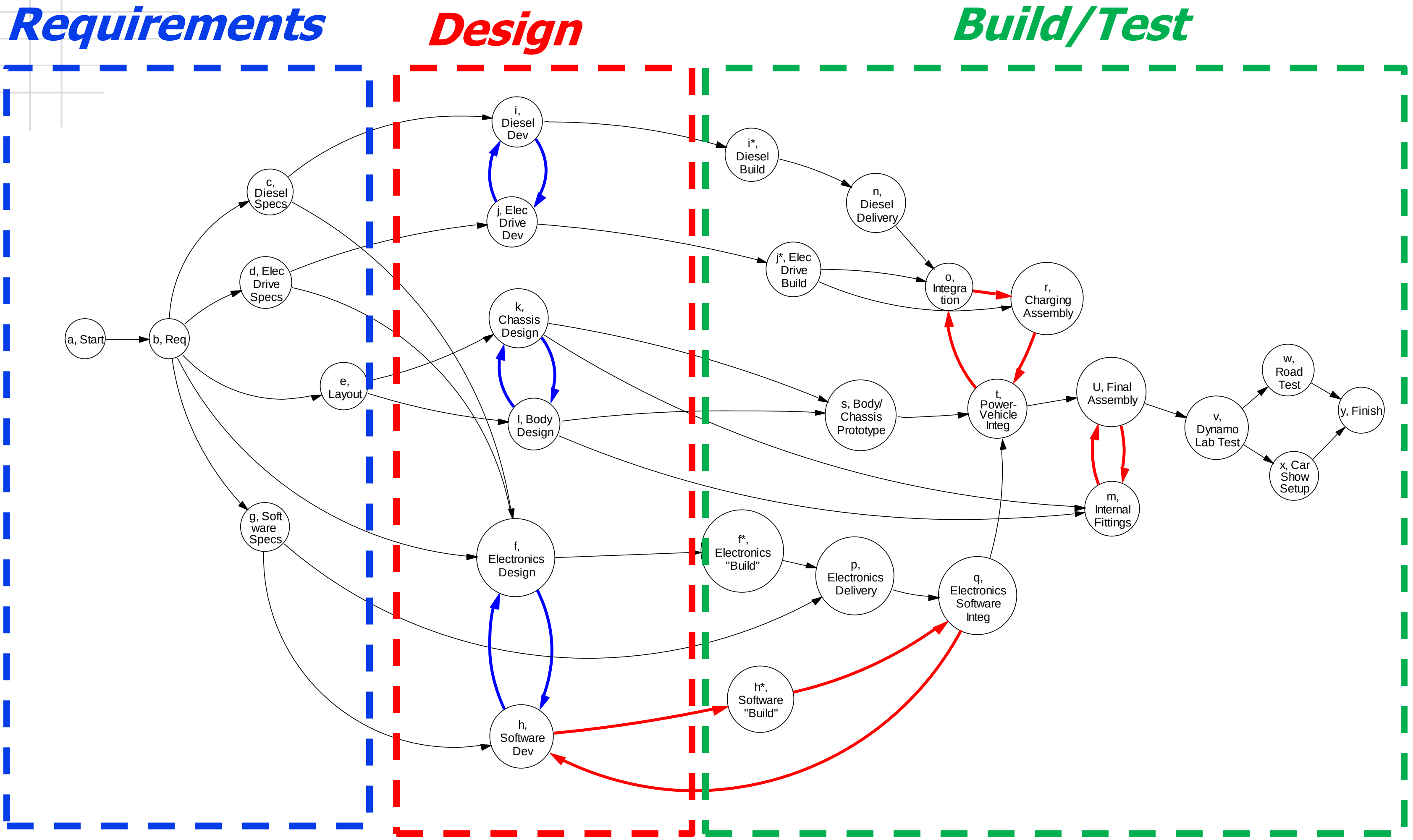


	Requirements Definition				
a	Start		1/1/10		
b	Requirements	60	1/1/10	3/25/10	a
c	Diesel Specs	30	3/25/10	5/6/10	b
d	Elec Drive Specs	30	3/26/10	5/6/10	b
g	Software Specs	72	3/26/10	7/5/10	b
	Body & Chassis				
e	Layout	40	3/26/10	6/1/10	b
k	Chassis Design	102	6/2/10	10/21/10	e
l	Body Design	90	6/2/10	10/5/10	e
s	Body Chassis Prototype	48	10/22/10	12/28/10	k,l
	Electronics & Software				
f	Electronics Design	90	5/7/10	9/9/10	b,c,d
h	Software Dev	150	9/10/10	4/7/11	g,f
p	Electronics Delivery	72	9/10/10	12/20/10	g,f
q	Electronics Software Int	30	4/8/11	5/19/11	h,p
	Diesel & Electric Drive				
i	Diesel Dev	180	5/7/10	1/13/11	c
j	Elec Drive Dev	90	5/7/10	9/9/10	d
n	Diesel Delivery & Check	12	1/14/11	1/31/11	i
	Vehicle & Power Train Int				
o	Integration	60	2/1/11	4/25/11	j,n
r	Charging Assembly	12	4/26/11	5/11/11	j,o
t	Power Vehicle Int	60	5/20/11	8/11/11	q,r,s
	Final Assembly				
m	Internal Fittings	48	10/22/10	12/28/10	k,l
u	Final Assembly	30	8/12/11	9/22/11	m,t
	Testing & Completion				
v	Dynamo Lab Test	30	9/23/11	11/3/11	u
w	Road Test	72	11/4/11	2/13/12	v
x	Car Show Setup	45	11/4/11	1/5/12	v
y	Finish			2/13/12	

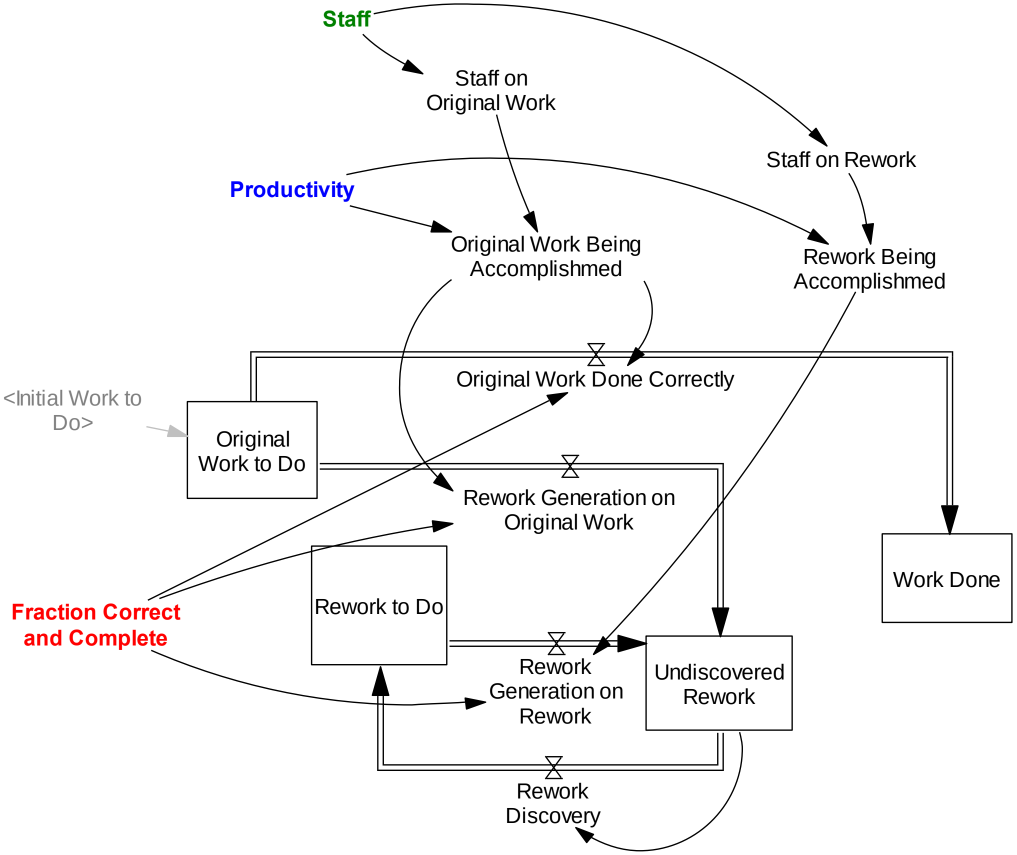
Observation/ Question: no coupling between major design phases?

Massachusetts Institute of Technology

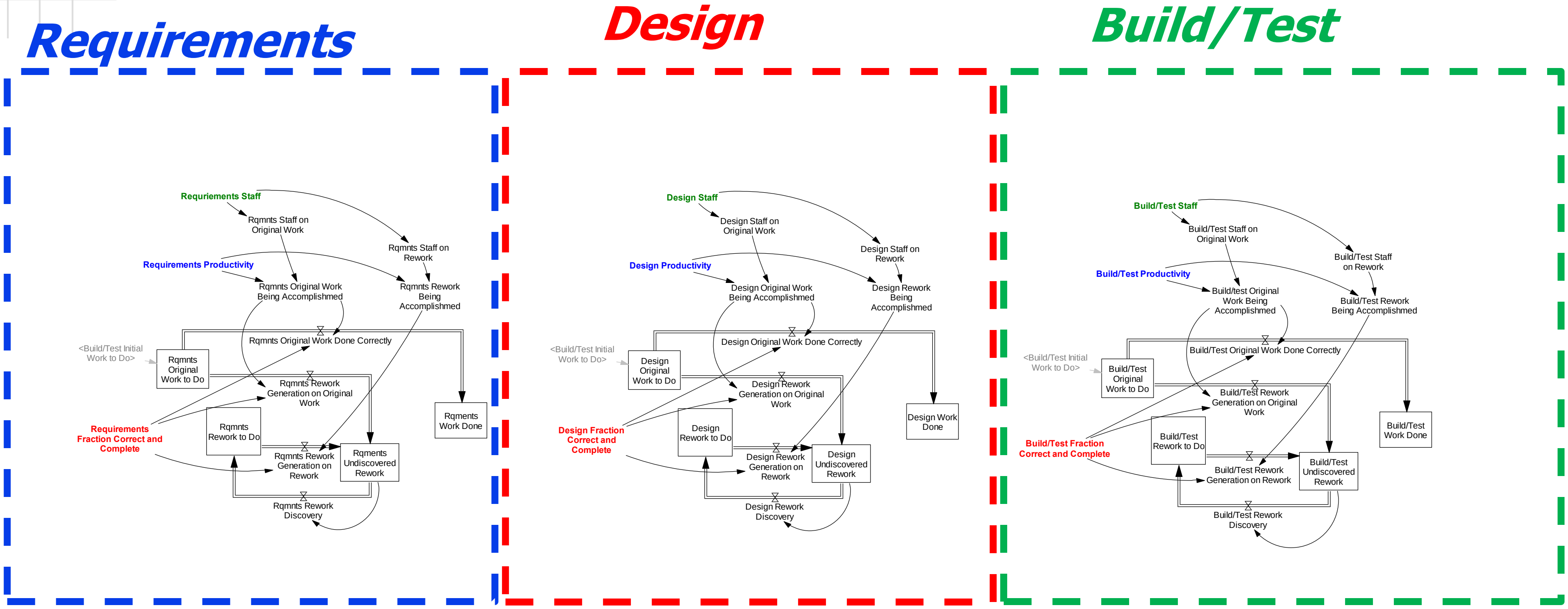
Network Diagram With “Build” Steps Added and Major Work Phases Highlighted



In an SD Model, Each Phase of Work Could Be Represented by a Rework Cycle


Staff
Staff on Original Work
Staff on Rework
Productivity
Original Work Being Accomplishmed
Rework Being Accomplishmed
Original Work Done Correctly
<Initial Work to Do>
Original Work to Do
Rework Generation on Original Work
Work Done
Fraction Correct and Complete
Rework to Do
Rework Generation on Rework
Undiscovered Rework
Rework Discovery

… Expanded to Three Phases


Requirements
Design
Build/Test
Requriements Staff
Rqmnts Staff on Original Work
Rqmnts Staff on Rework
Requirements Productivity
Rqmnts Original Work Being Accomplishmed
Rqmnts Rework Being Accomplishmed
Rqmnts Original Work Done Correctly
<Build/Test Initial Work to Do>
Rqmnts Original Work to Do
Rqmnts Rework Generation on Original Work
Rqments Work Done
Requirements Fraction Correct and Complete
Rqmnts Rework to Do
Rqmnts Rework Generation on Rework
Rqments Undiscovered Rework
Rqmnts Rework Discovery
Design Staff
Design Staff on Original Work
Design Staff on Rework
Design Productivity
Design Original Work Being Accomplishmed
Design Rework Being Accomplishmed
Design Original Work Done Correctly
<Build/Test Initial Work to Do>
Design Original Work to Do
Design Rework Generation on Original Work
Design Work Done
Design Fraction Correct and Complete
Design Rework to Do
Design Rework Generation on Rework
Design Undiscovered Rework
Design Rework Discovery
Build/Test Staff
Build/Test Staff on Original Work
Build/Test Staff on Rework
Build/Test Productivity
Build/test Original Work Being Accomplishmed
Build/Test Rework Being Accomplishmed
Build/Test Original Work Done Correctly
<Build/Test Initial Work to Do>
Build/Test Original Work to Do
Build/Test Rework Generation on Original Work
Build/Test Work Done
Build/Test Fraction Correct and Complete
Build/Test Rework to Do
Build/Test Rework Generation on Rework
Build/Test Undiscovered Rework
Build/Test Rework Discovery

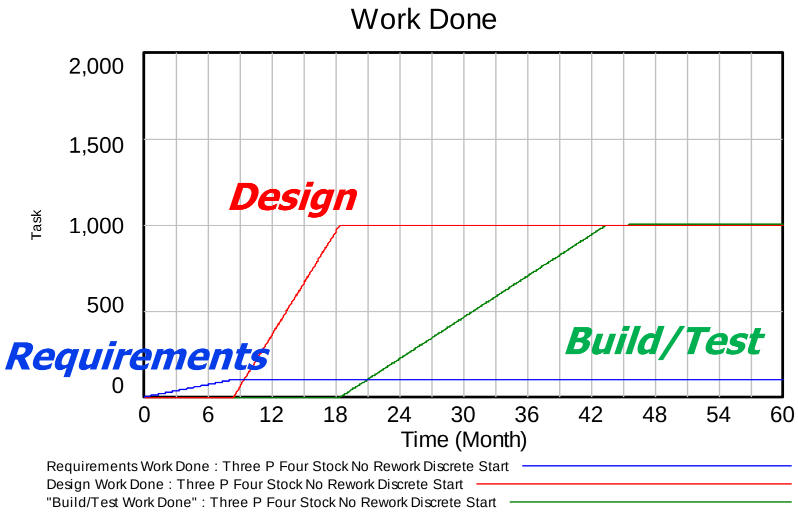
Assumptions:

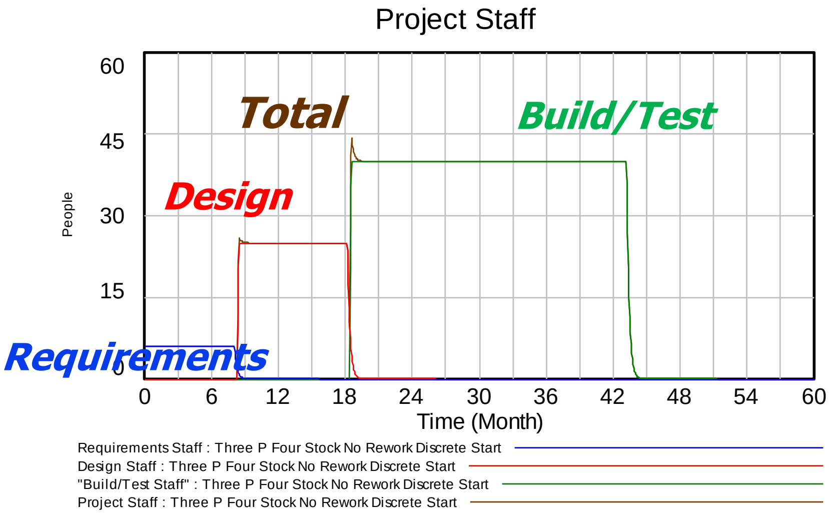
Scope = 100 Tasks
Staff = 4,
Productivity = 2 tasks/month/person
Duration = 12.5 months

Scope = 1000 tasks
Staff = 20
Productivity = 4 tasks/month/person
Duration = 12.5 months

Scope = 1000 tasks
Staff = 40
Productivity = 1 tasks/month/person
Duration = 25 months

Simulation of project assuming no rework …





Sources of Rework -- Categories

1. Classical “Quality” or design mis-execution.
2. Technical complexity/uncertainty; customer uncertainty.
3. Work done “correctly” but ultimately needing rework because it builds on

Classical “Quality”

- There are two (or more) important sources of these errors –
  - “People” factors such as fatigue, inexperience, skill mismatches, etc.
  - “Coupled” tasks that require shared information but which are done independently
  - Organizational size & complexity?
- ***These types of errors can in theory be discovered by further “design” work, such as doing downstream design work, QA, design reviews, planned iteration, etc.***

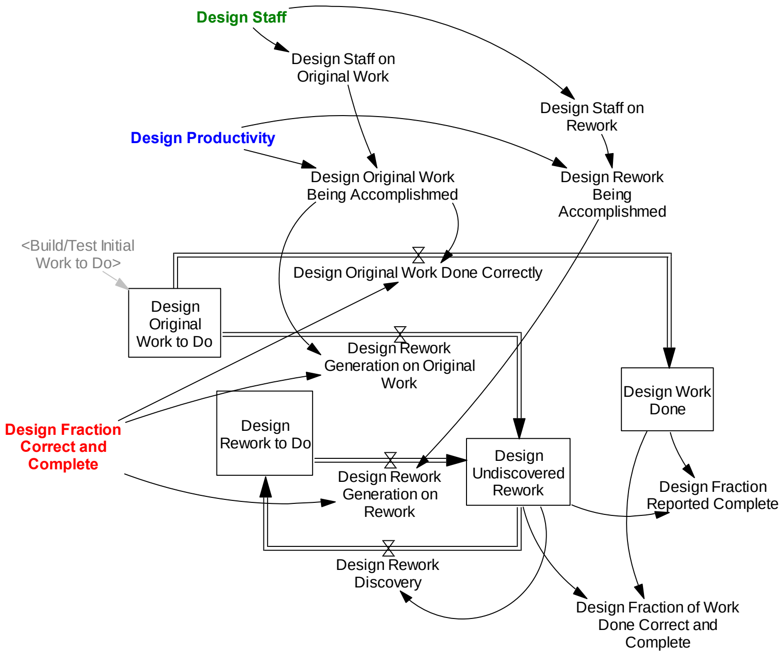
Technical complexity/uncertainty; customer uncertainty.

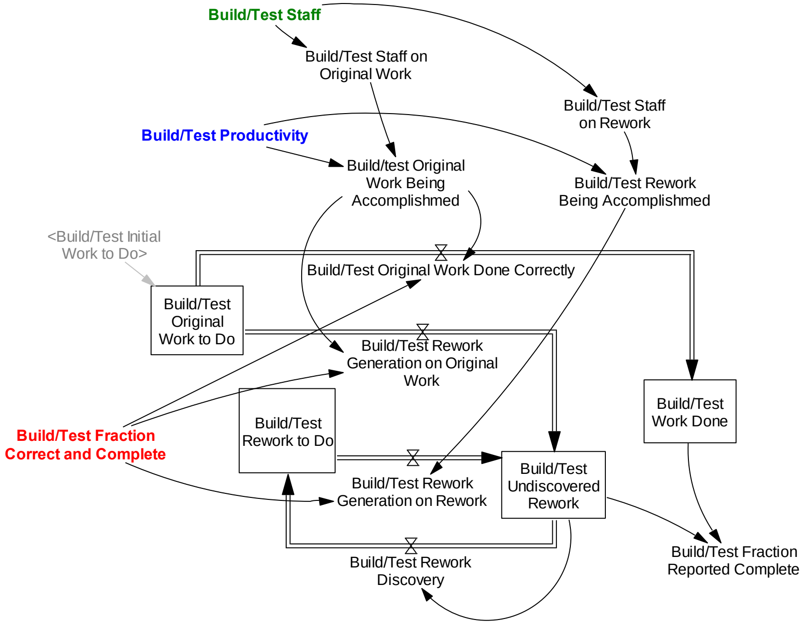
- Technical: “novelty” of the project
- Customer: Same logic might apply to novel/new “uncertain customer requirements”? I.e., where say a software product is being developed and the customer may not know what they want until they see it in practice.]
- ***These types of errors can only be discovered by doing build and test work. Fixing these errors may just require redoing tasks, but may also require new tasks to make the technology work (e.g., and additional controller).***

“Knock-on” Rework

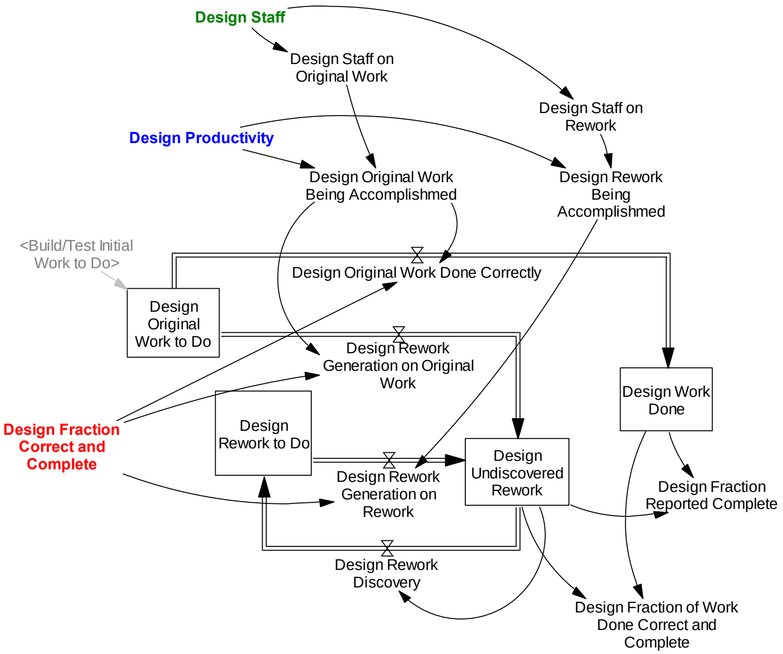
- Work done “correctly” but ultimately needing rework because it builds on
  - Incorrect prior work
  - Assumptions about technology or customer requirements which prove incorrect.
  - Exogenous changes in requirements, scope, etc.
- ***These errors could be discovered in design or build/test, depending on the source.***

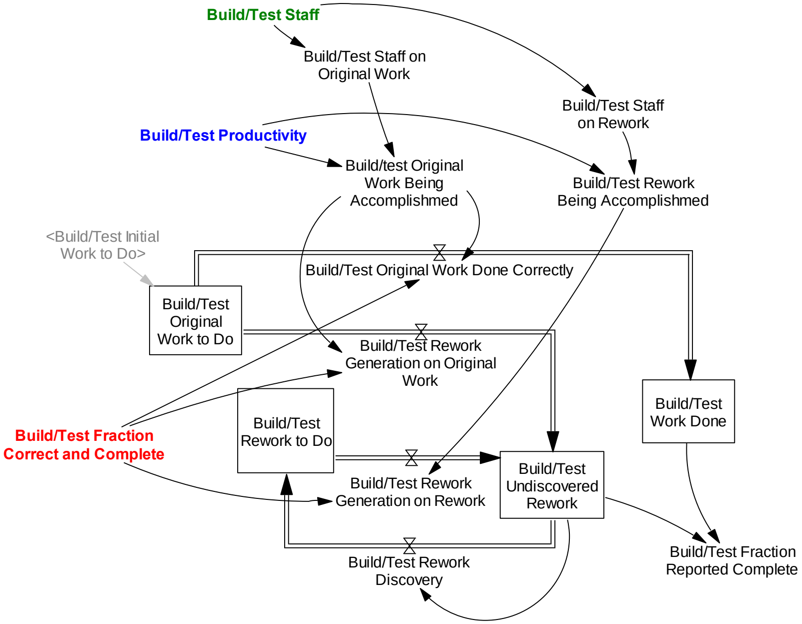
How does Design Affect Build?


Design Staff
Design Staff on Original Work
Design Staff on Rework
Design Productivity
Design Original Work Being Accomplishmed
Design Rework Being Accomplishmed
<Build/Test Initial Work to Do>
Design Original Work to Do
Design Original Work Done Correctly
Design Rework Generation on Original Work
Design Work Done
Design Fraction Correct and Complete
Design Rework to Do
Design Undiscovered Rework
Design Rework Generation on Rework
Design Fraction Reported Complete
Design Rework Discovery
Design Fraction of Work Done Correct and Complete

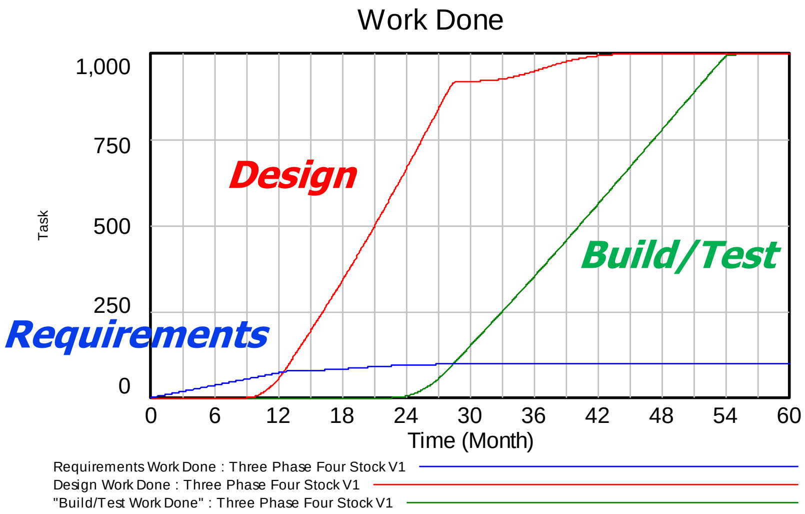

Build/Test Staff
Build/Test Staff on Original Work
Build/Test Staff on Rework
Build/Test Productivity
Build/test Original Work Being Accomplishmed
Build/Test Rework Being Accomplishmed
<Build/Test Initial Work to Do>
Build/Test Original Work to Do
Build/Test Original Work Done Correctly
Build/Test Rework Generation on Original Work
Build/Test Work Done
Build/Test Fraction Correct and Complete
Build/Test Rework to Do
Build/Test Undiscovered Rework
Build/Test Rework Generation on Rework
Build/Test Fraction Reported Complete
Build/Test Rework Discovery

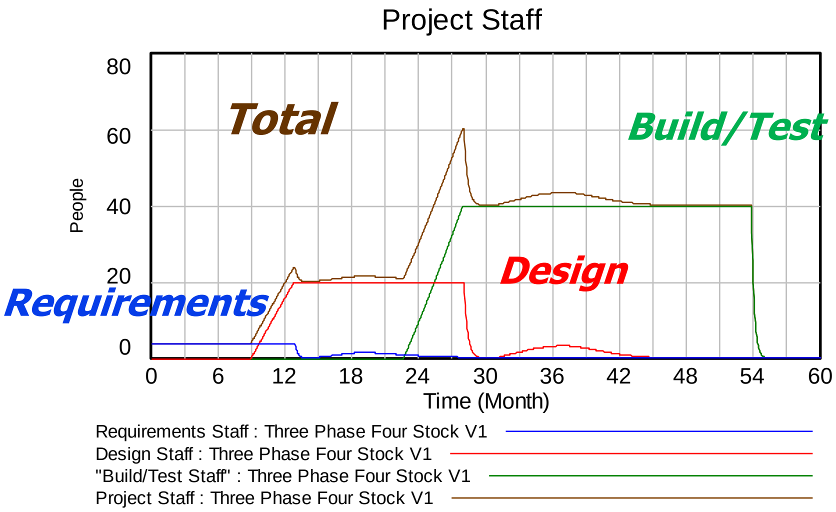
How is Design Rework Discovered?


Design Staff
Design Staff on Original Work
Design Staff on Rework
Design Productivity
Design Original Work Being Accomplishmed
Design Rework Being Accomplishmed
<Build/Test Initial Work to Do>
Design Original Work Done Correctly
Design Original Work to Do
Design Rework Generation on Original Work
Design Work Done
Design Fraction Correct and Complete
Design Rework to Do
Design Undiscovered Rework
Design Rework Generation on Rework
Design Fraction Reported Complete
Design Rework Discovery
Design Fraction of Work Done Correct and Complete

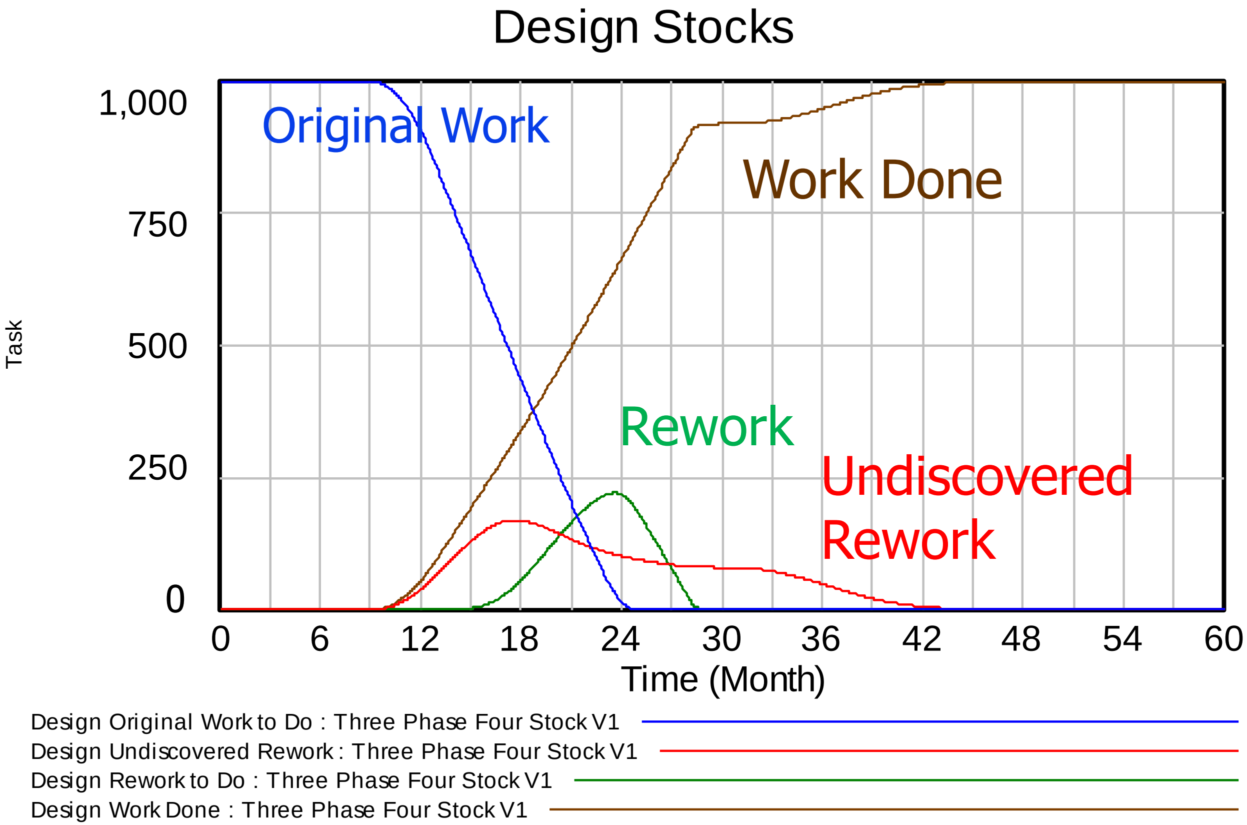

Build/Test Staff
Build/Test Staff on Original Work
Build/Test Staff on Rework
Build/Test Productivity
Build/test Original Work Being Accomplishmed
Build/Test Rework Being Accomplishmed
<Build/Test Initial Work to Do>
Build/Test Original Work Done Correctly
Build/Test Original Work to Do
Build/Test Rework Generation on Original Work
Build/Test Work Done
Build/Test Fraction Correct and Complete
Build/Test Rework to Do
Build/Test Undiscovered Rework
Build/Test Rework Generation on Rework
Build/Test Fraction Reported Complete
Build/Test Rework Discovery

Simulation Incorporating Rework and Rework Discovery

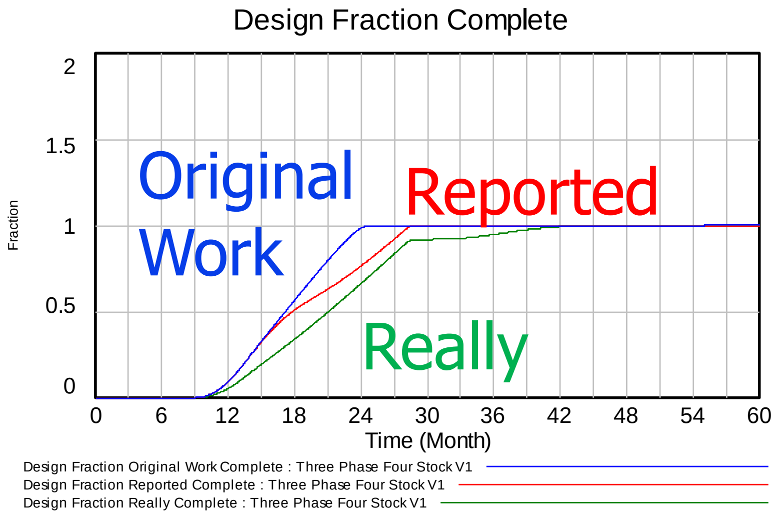


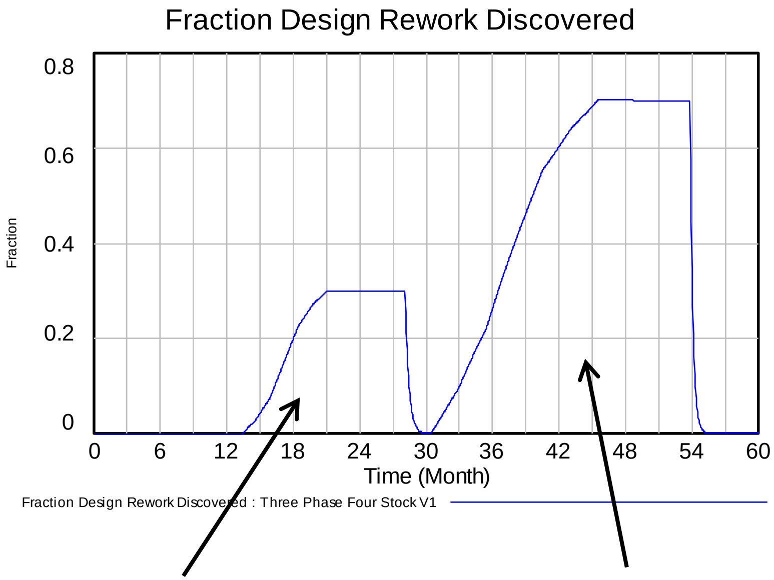


Notes: (1) Details of simulation do not correspond to NMM Case; (2) so far, we do not represent project control actions and effects, so staff/hours worked are constant.


Design Stocks
1,000
750
500
250
0
Task
Original Work
Work Done
Rework
Undiscovered Rework
0
6
12
18
24
30
36
42
48
54
60
Time (Month)
Design Original Work to Do : Three Phase Four Stock V1
Design Undiscovered Rework : Three Phase Four Stock V1
Design Rework to Do : Three Phase Four Stock V1
Design Work Done : Three Phase Four Stock V1

Progress and Rework Discovery



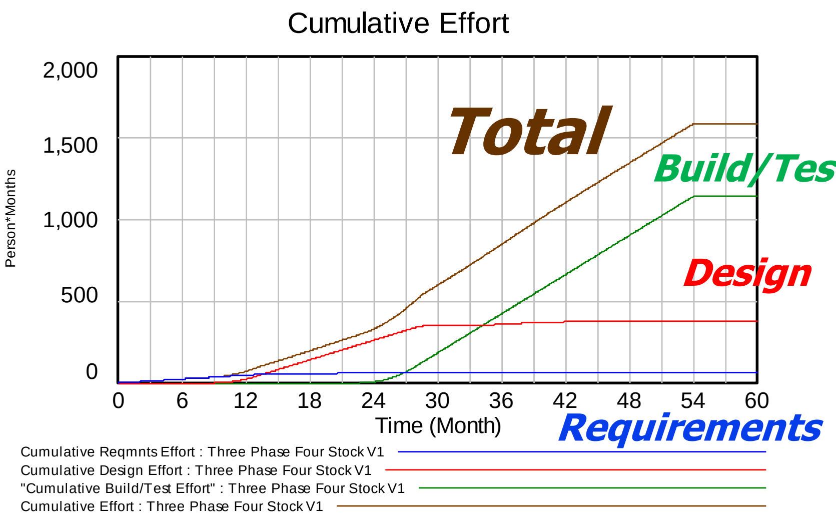


Discovery by design

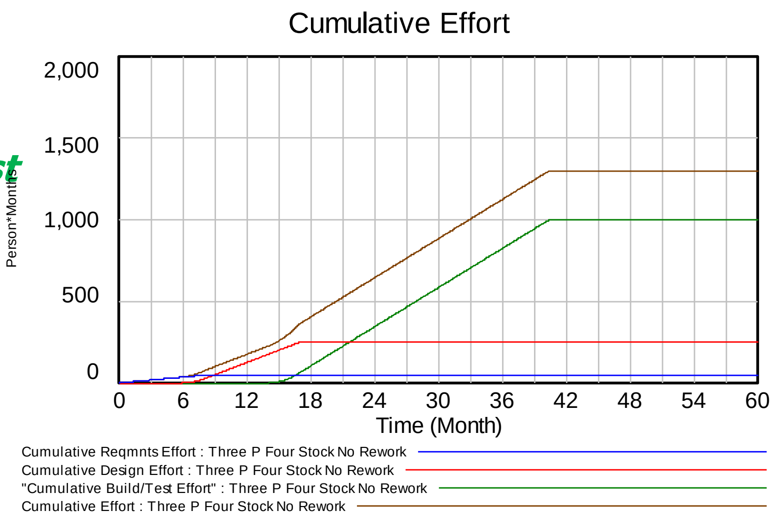
Discovery by build

Rework Increases Cost and Delays Finish

With Rework

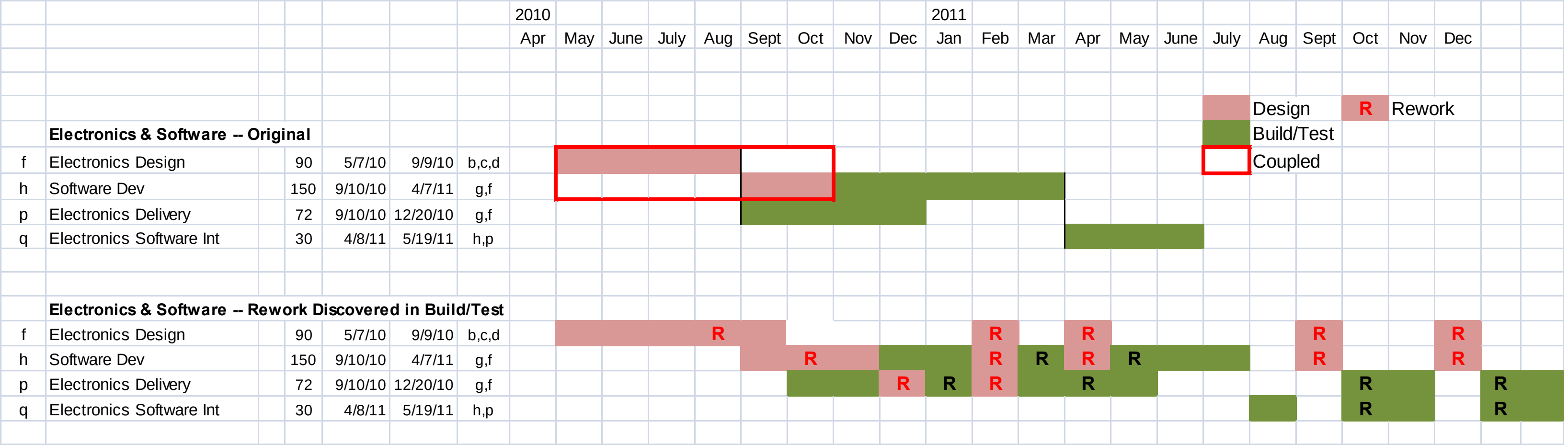


Without Rework



Revised Network/Gantt to reflect rework discovery (by build, not design)

Insert some Rework Tasks in Design, more in Build/Test



🟥 Design R Rework
🟩 Build/Test
□ Coupled

						2010									2011													
						Apr	May	June	July	Aug	Sept	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	June	July	Aug	Sept	Oct	Nov	Dec		
	Electronics & Software -- Original																											
f	Electronics Design	90	5/7/10	9/9/10	b,c,d		🟥	🟥	🟥	🟥																		
h	Software Dev	150	9/10/10	4/7/11	g,f						🟥	🟥	🟩	🟩	🟩	🟩	🟩											
p	Electronics Delivery	72	9/10/10	12/20/10	g,f						🟩	🟩	🟩	🟩														
q	Electronics Software Int	30	4/8/11	5/19/11	h,p													🟩	🟩	🟩								
	Electronics & Software -- Rework Discovered in Build/Test																											
f	Electronics Design	90	5/7/10	9/9/10	b,c,d		🟥	🟥	🟥	R	🟥					R		R					R			R		
h	Software Dev	150	9/10/10	4/7/11	g,f						🟥	R	🟥	🟩	🟩	R	R	R	R	🟩	🟩		R			R		
p	Electronics Delivery	72	9/10/10	12/20/10	g,f							🟩	🟩	R	R	R	🟩	R	🟩					R	🟩		R	🟩
q	Electronics Software Int	30	4/8/11	5/19/11	h,p																	🟩		R	🟩		R	🟩

Management Issues

- Planned Iterations (vs dealing with unplanned iterations later) – How many iterations?
- When to start build phases – how much overlap with design phases.
- Factors to Consider:
  - Type of project (determines amount of rework discoverable in design, amount of rework, number of tasks which must be repeated for each iteration)
  - Relative cost of design vs build/test (determines the relative cost of spending more in design)

More on these later in term …

Role of Tools

- SD:
  - Evaluate the macro tradeoffs in terms of impact on cost and schedule
  - Coming up with a good staffing plan
- DSM:
  - Determine where design iterations are needed and how many tasks would need to be repeated per iteration
- Network:
  - Guide operational task planning and resource scheduling

Dynamics of Project Performance

The “rework cycle”

- Fraction correct and complete ✓
- Undiscovered rework ✓

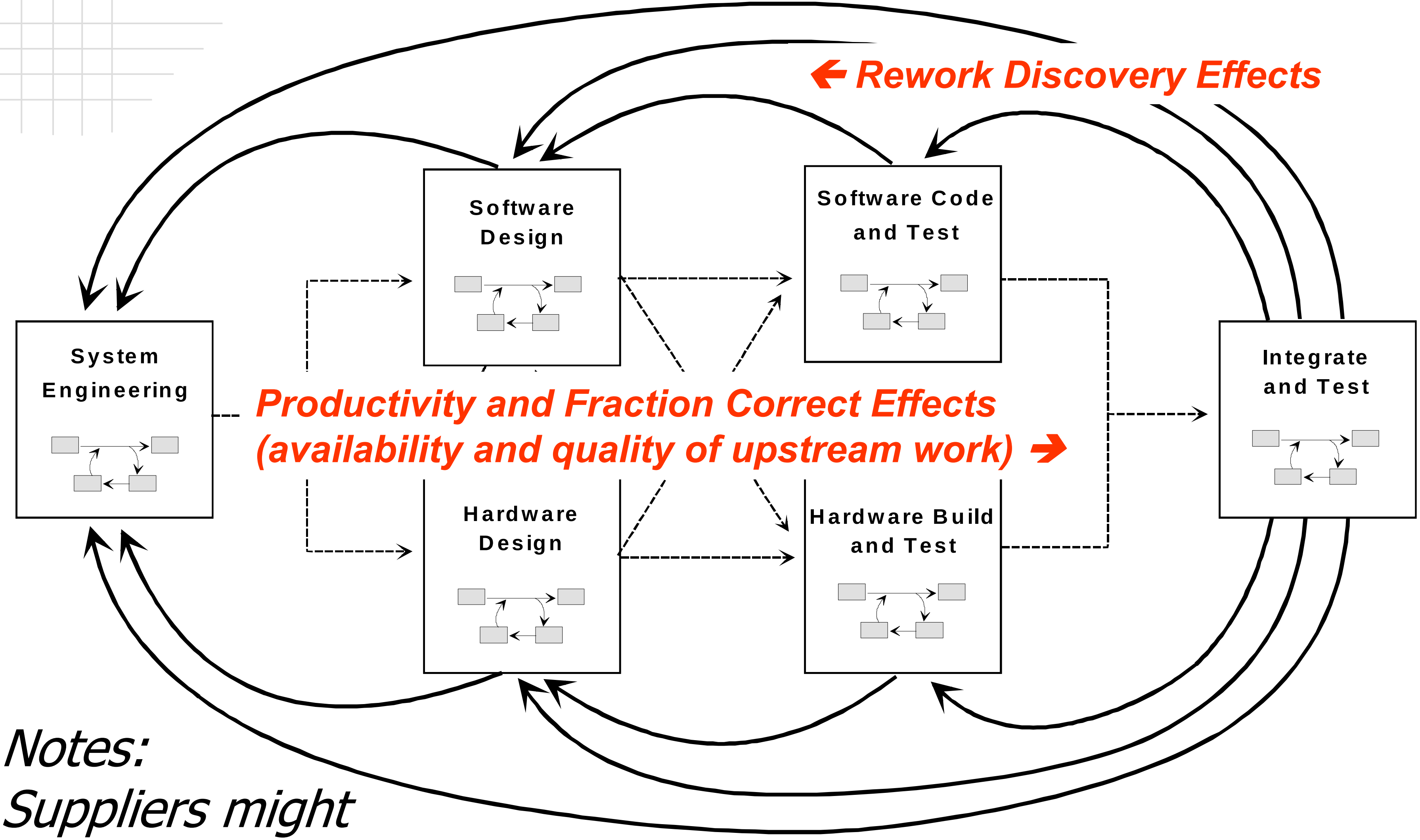
Feedback effects on productivity and fraction correct (Next class)

- Negative, controlling
- Positive, re-enforcing, often “vicious circles”

Knock-on effects between work phases

- Availability and quality of work products ✓
- Progress to discover upstream rework ✓

Knock-on Between Phases: A system dynamics model usually represents several phases of work …



🡐 Rework Discovery Effects
Software Design
Software Code and Test
System Engineering
Integrate and Test
Productivity and Fraction Correct Effects (availability and quality of upstream work) 🡒
Hardware Design
Hardware Build and Test

Notes:
Suppliers might be a "phase"

Why Represent Separate Phases?

- Correspond to gate reviews and deliverables
- Different units of work (and therefore data)
- Different types of labor
- Disproportionate increase in costs of fixing errors as move downstream
- Address issues of overlap/concurrency
- Opportunity to adjust plans, delay start, between phases

Today's Agenda

- Overview: Causes of Project Dynamics
- The Rework Cycle
- Integrating Tools in Project Planning
- ➡ Simple Model of Project Dynamics, Pt. 1

Hard tools force us to be more explicit, and accurately simulate the consequences of our models ...

"Soft" tools --

- behavior-over-time graphs
- cause-effect diagramming
- mental simulation

Tools for describing dynamics

"Hard" tools --

- computer models
- computer simulation
- calibration to data
- sensitivity and what-if analyses

Tools for quantifying dynamics

We will use two models …

- Simple rework cycle model with project control feedbacks
  - HW#3 – develop simple model without feedbacks
  - Feedbacks added in class, given in HW#5
- Full rework cycle model with two phases of work
  - No project control feedback
  - Model given to you for HW#3 and HW#5

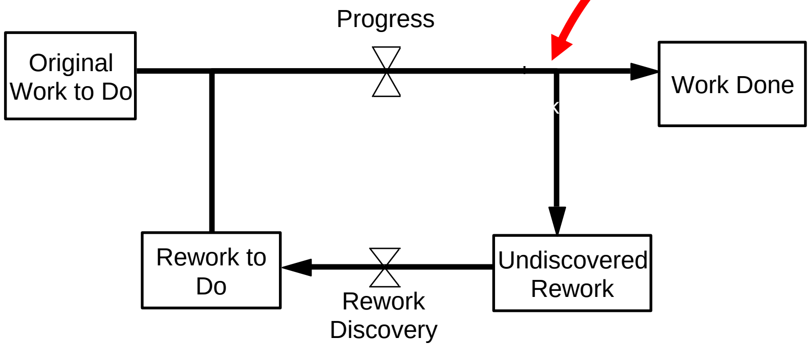
Development of "Simple" Project Model 1

- Rework cycle model (HW#3)
  - Three stocks
  - Variable rework discovery time
  - "Errors Build Errors" Feedback
- Project control & Side Effects (HW#5)
  - Work Intensity/Schedule Pressure & "Haste Makes Waste"
  - Staffing & Experience Dilution
  - Slip Schedule

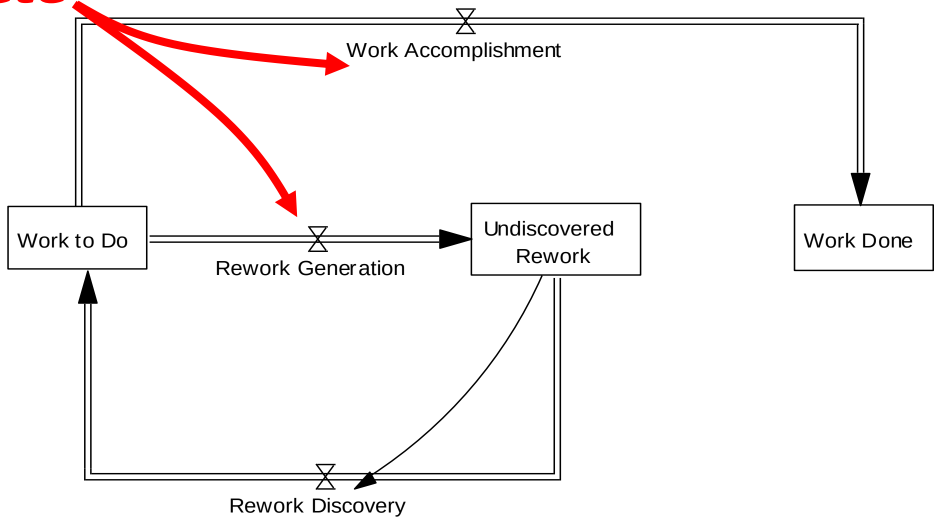
Two Views of the Rework Cycle

Fraction Correct & Complete

Complete Model

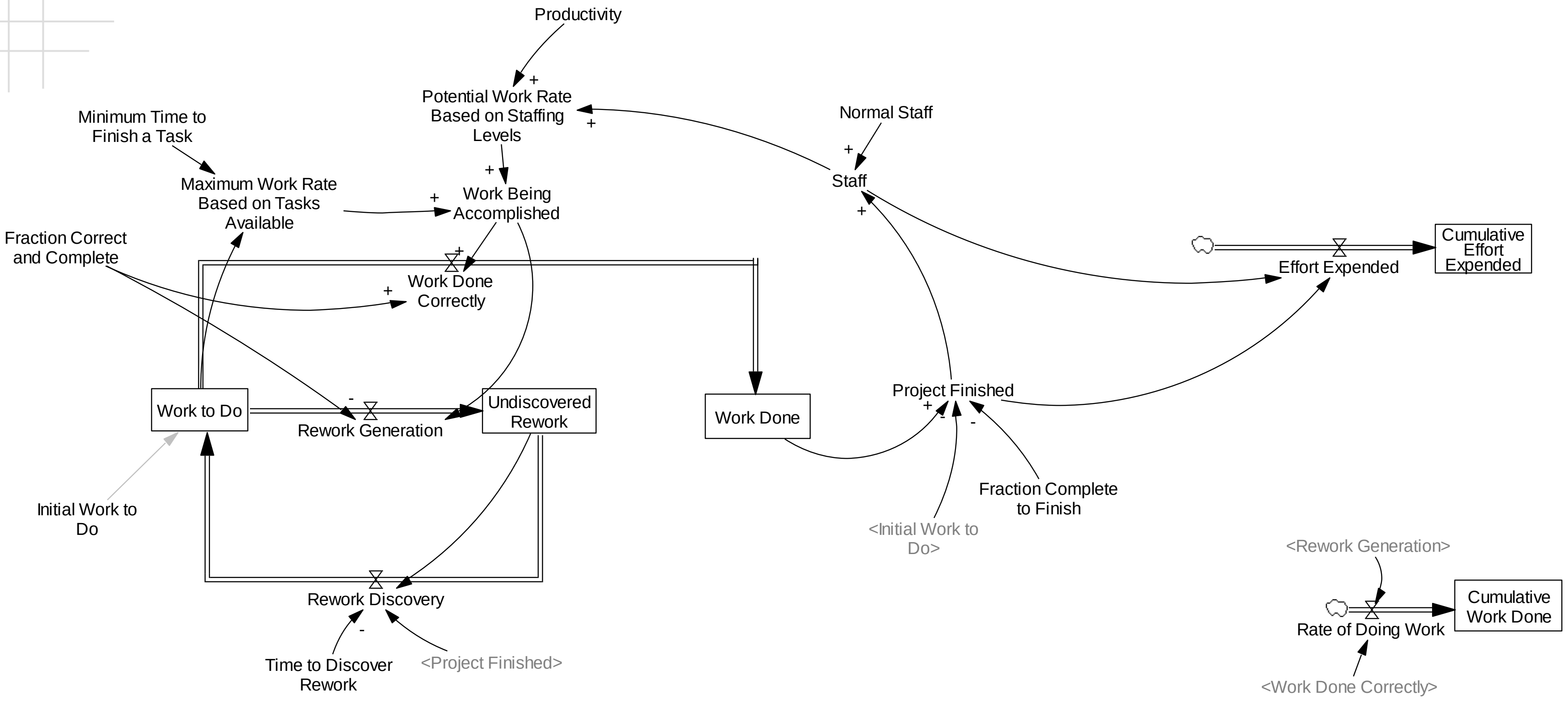


Simplified Version



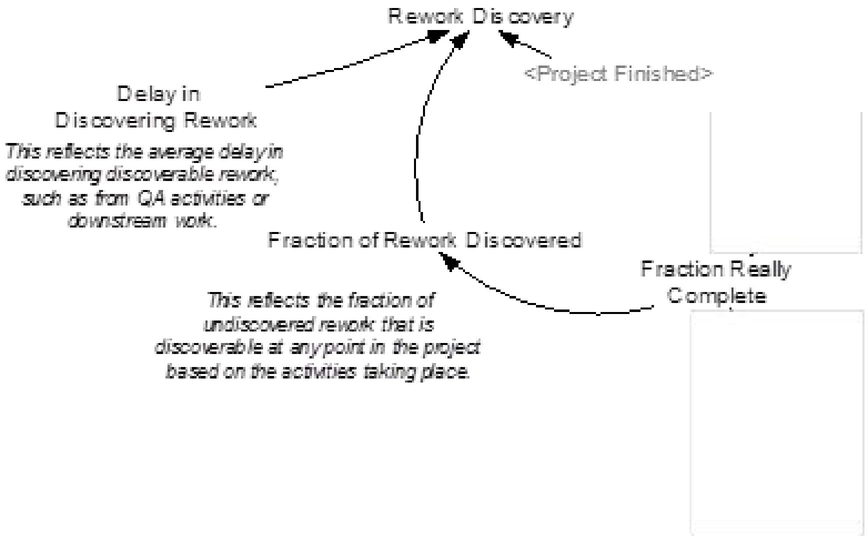
The simplified version assumes that rework tasks require the same effort as original tasks, and that it is not important to distinguish between original work and rework.

Simple Rework Cycle Model



Steps 1-4 of Homework #3

Rework Discovery Depends on Progress



Rework Discovery

<Project Finished>

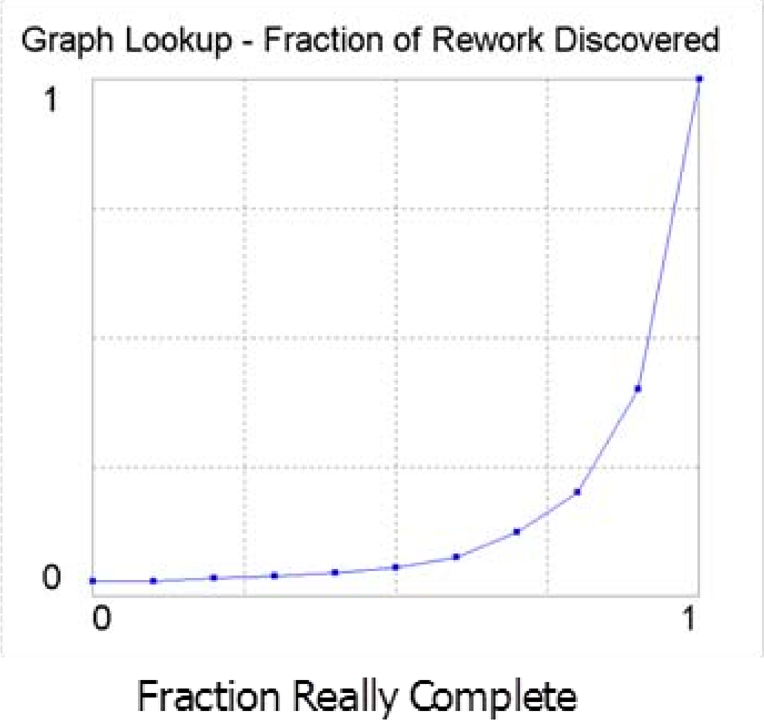
Delay in Discovering Rework

This reflects the average delay in discovering discoverable rework, such as from QA activities or downstream work.

Fraction of Rework Discovered

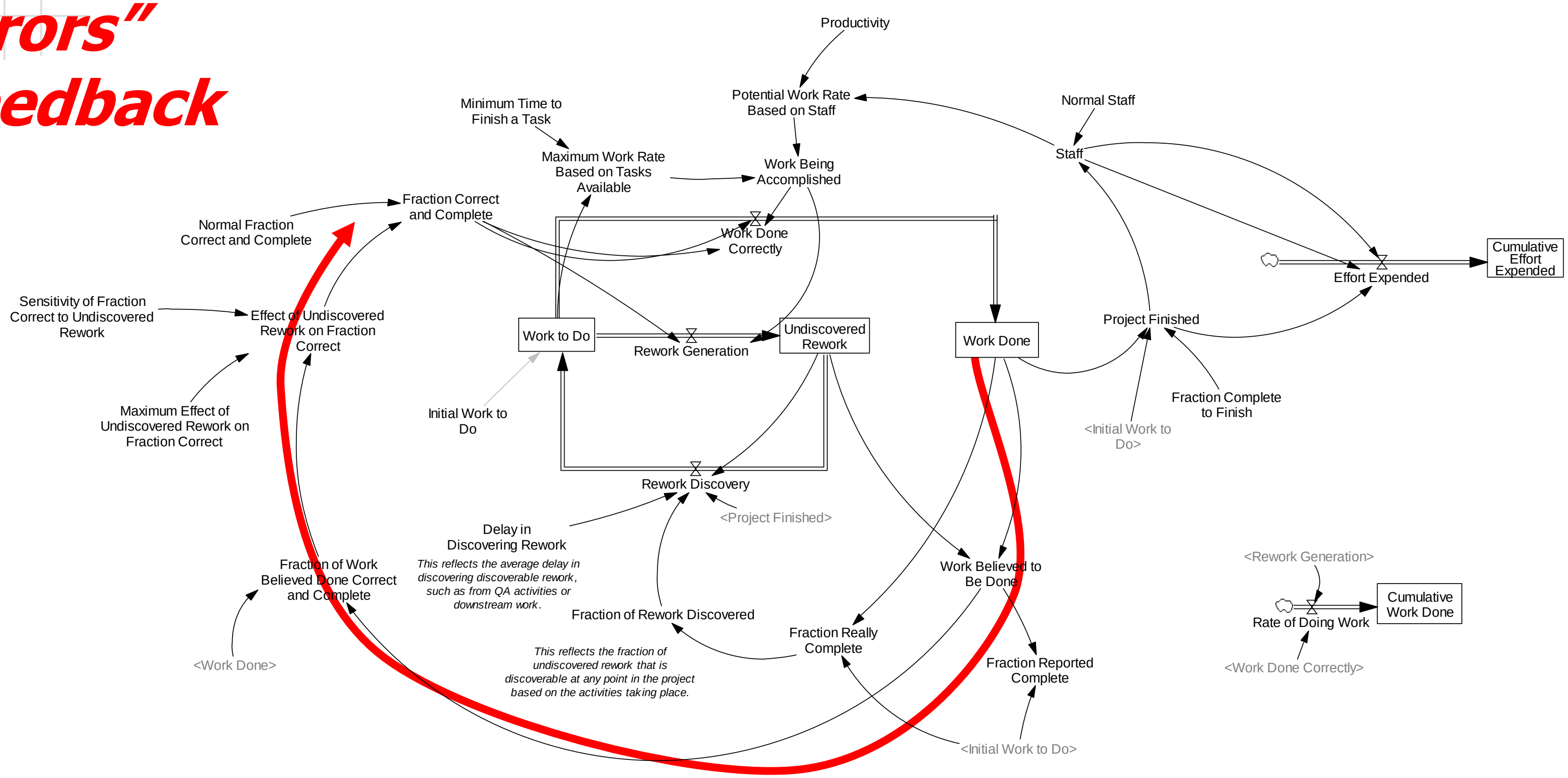
Fraction Really Complete

This reflects the fraction of undiscovered rework that is discoverable at any point in the project based on the activities taking place.



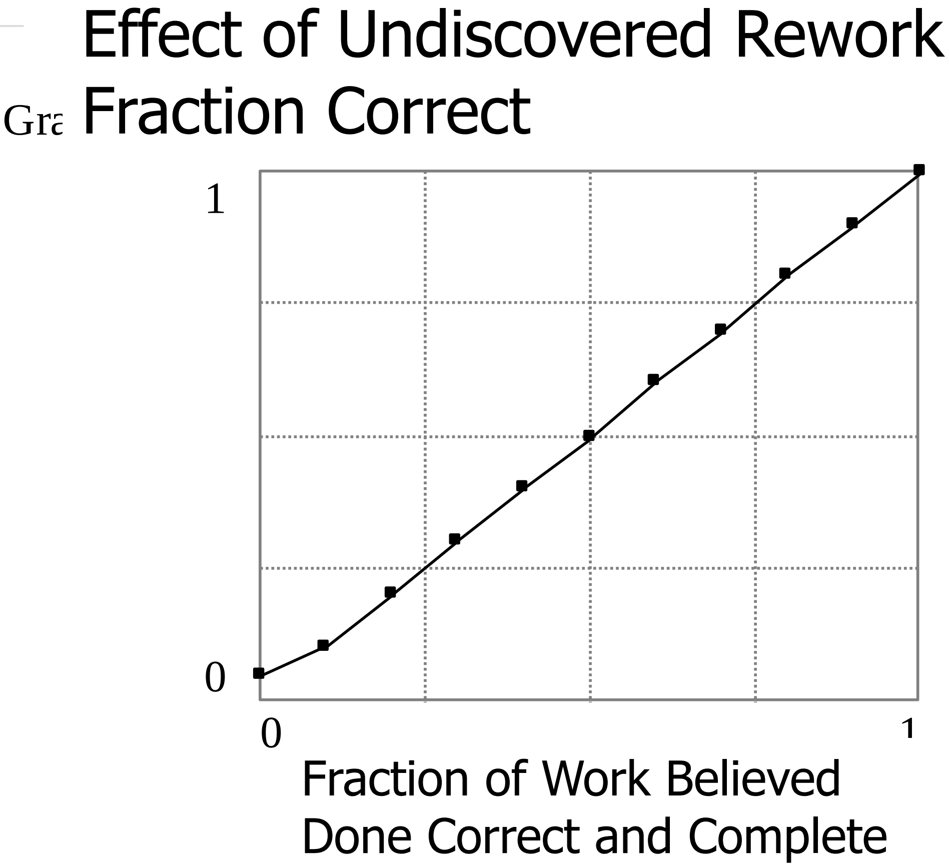
Complete Simple Model 1

"Errors on Errors" Feedback


Productivity
Potential Work Rate Based on Staff
Normal Staff
Staff
Minimum Time to Finish a Task
Maximum Work Rate Based on Tasks Available
Work Being Accomplished
Normal Fraction Correct and Complete
Fraction Correct and Complete
Work Done Correctly
Cumulative Effort Expended
Effort Expended
Sensitivity of Fraction Correct to Undiscovered Rework
Effect of Undiscovered Rework on Fraction Correct
Work to Do
Rework Generation
Undiscovered Rework
Work Done
Project Finished
Maximum Effect of Undiscovered Rework on Fraction Correct
Initial Work to Do
Fraction Complete to Finish
<Initial Work to Do>
Rework Discovery
<Project Finished>
Delay in Discovering Rework
This reflects the average delay in discovering discoverable rework, such as from QA activities or downstream work.
Fraction of Work Believed Done Correct and Complete
Work Believed to Be Done
<Rework Generation>
Cumulative Work Done
Rate of Doing Work
Fraction of Rework Discovered
Fraction Really Complete
Fraction Reported Complete
<Work Done>
This reflects the fraction of undiscovered rework that is discoverable at any point in the project based on the activities taking place.
<Work Done Correctly>
<Initial Work to Do>

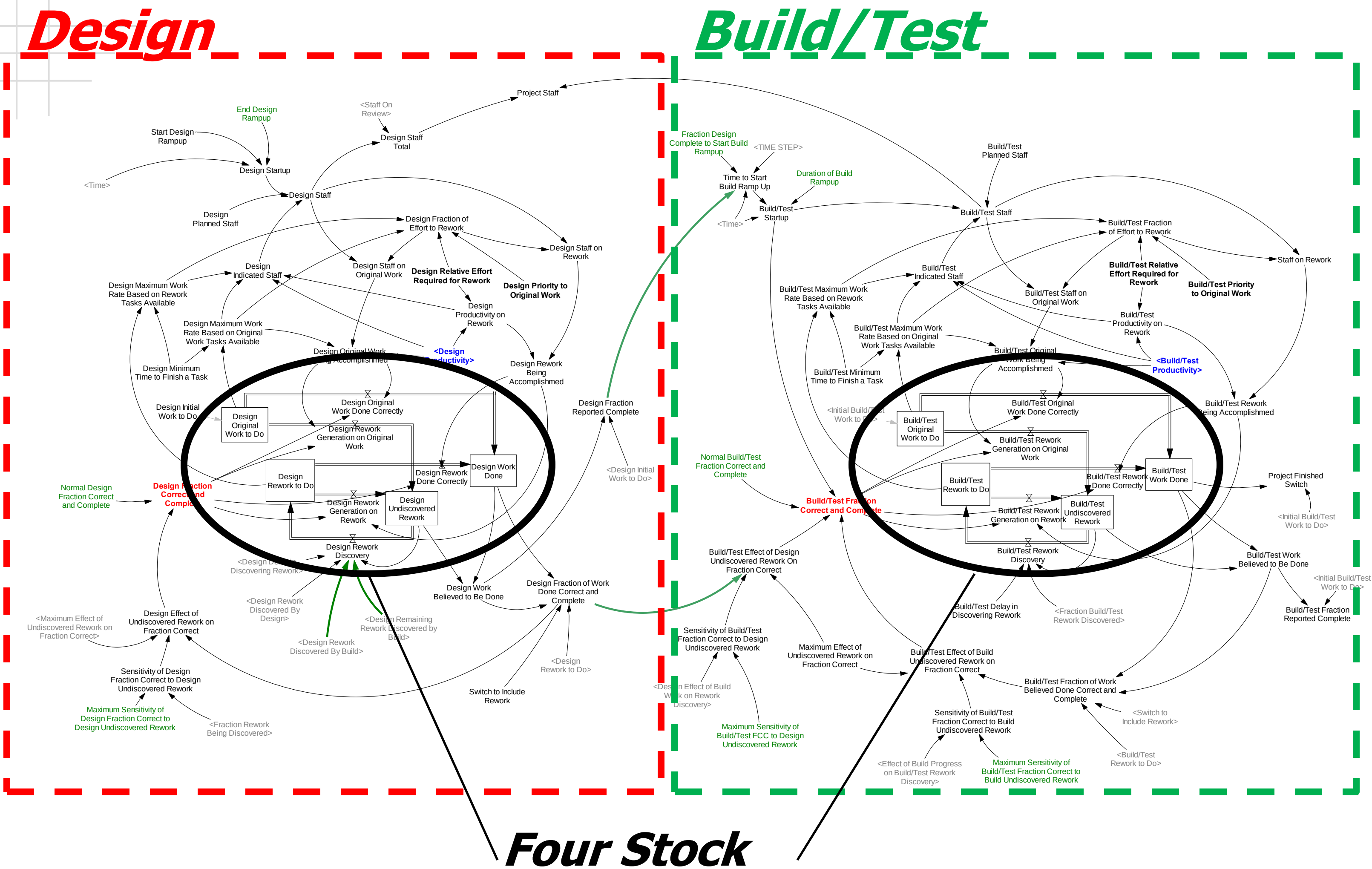
Massachusetts Institute of Technology

Effect of Undiscovered Rework on Fraction Correct:



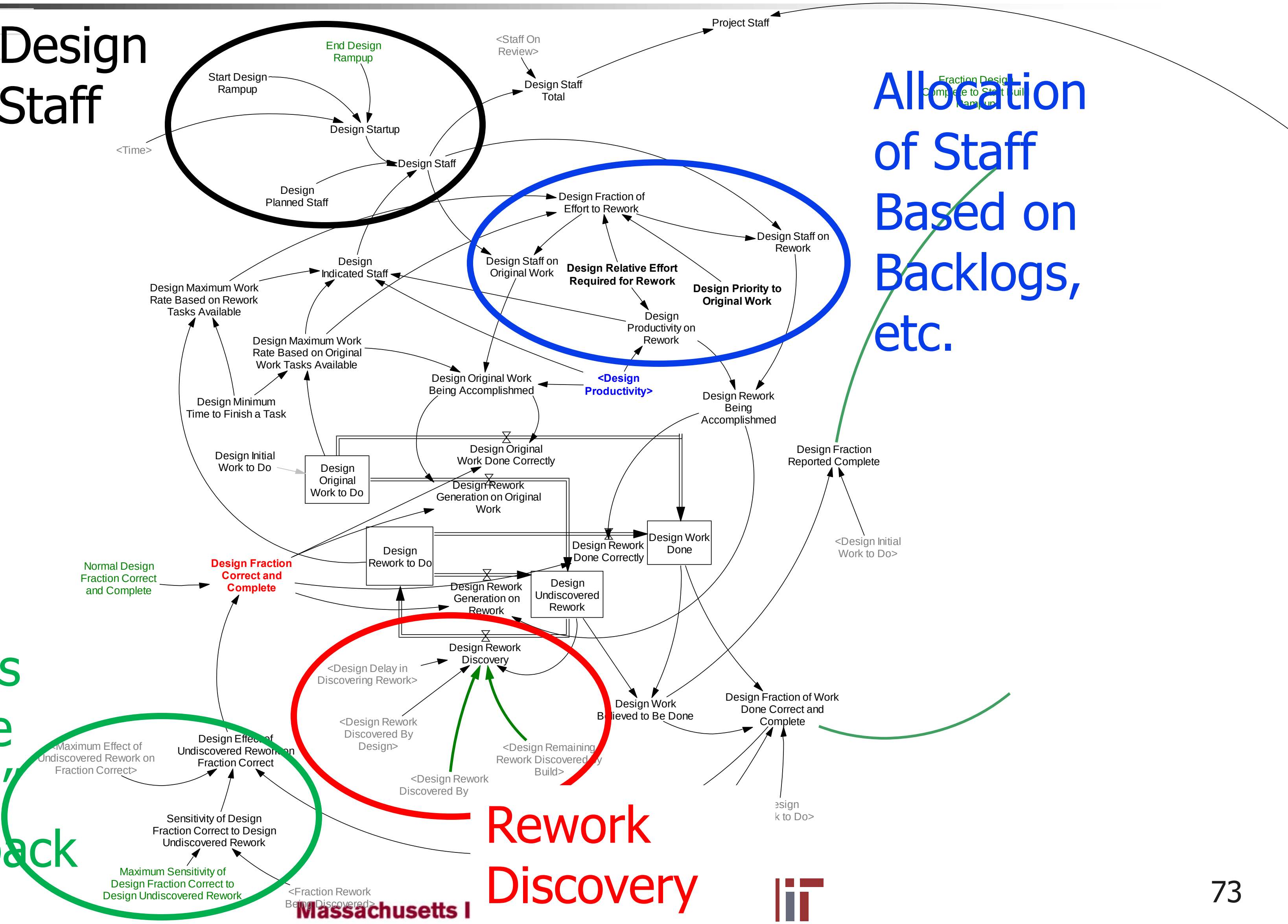
Note: The effect of undiscovered rework on fraction correct is assumed to be proportional -- an error in past work creates an error in current work. Given that in this simple model fraction correct represents several effects of work errors, this strong relationship may be reasonable.

Work Flows & Staffing in "Simple" Two Phase Model


Design
Build/Test
Project Staff
<Staff On Review>
End Design Rampup
Start Design Rampup
Design Staff Total
<Time>
Design Startup
Design Staff
Design Planned Staff
Design Fraction of Effort to Rework
Design Staff on Rework
Design Indicated Staff
Design Staff on Original Work
Design Relative Effort Required for Rework
Design Priority to Original Work
Design Maximum Work Rate Based on Rework Tasks Available
Design Productivity on Rework
Design Maximum Work Rate Based on Original Work Tasks Available
Design Original Work Being Accomplished
<Design Productivity>
Design Rework Being Accomplished
Design Minimum Time to Finish a Task
Design Initial Work to Do
Design Original Work to Do
Design Original Work Done Correctly
Design Rework Generation on Original Work
Design Fraction Reported Complete
Design Work Done
Design Rework Done Correctly
Design Rework to Do
<Design Initial Work to Do>
Normal Design Fraction Correct and Complete
Design Fraction Correct and Complete
Design Rework Generation on Rework
Design Undiscovered Rework
Design Rework Discovery
<Design Delay in Discovering Rework>
Design Work Believed to Be Done
Design Fraction of Work Done Correct and Complete
<Design Rework Discovered By Design>
<Maximum Effect of Undiscovered Rework on Fraction Correct>
Design Effect of Undiscovered Rework on Fraction Correct
<Design Remaining Rework Discovered by Build>
<Design Rework Discovered By Build>
<Design Rework to Do>
Sensitivity of Design Fraction Correct to Design Undiscovered Rework
Switch to Include Rework
Maximum Sensitivity of Design Fraction Correct to Design Undiscovered Rework
<Fraction Rework Being Discovered>
Fraction Design Complete to Start Build Rampup
<TIME STEP>
Build/Test Planned Staff
Time to Start Build Ramp Up
Duration of Build Rampup
Build/Test Startup
<Time>
Build/Test Staff
Build/Test Fraction of Effort to Rework
Staff on Rework
Build/Test Indicated Staff
Build/Test Relative Effort Required for Rework
Build/Test Priority to Original Work
Build/Test Maximum Work Rate Based on Rework Tasks Available
Build/Test Staff on Original Work
Build/Test Productivity on Rework
Build/Test Maximum Work Rate Based on Original Work Tasks Available
Build/Test Original Work Being Accomplished
<Build/Test Productivity>
Build/Test Minimum Time to Finish a Task
<Initial Build/Test Work to Do>
Build/Test Original Work to Do
Build/Test Original Work Done Correctly
Build/Test Rework Being Accomplished
Build/Test Rework Generation on Original Work
Normal Build/Test Fraction Correct and Complete
Build/Test Rework to Do
Build/Test Rework Done Correctly
Build/Test Work Done
Project Finished Switch
Build/Test Fraction Correct and Complete
Build/Test Rework Generation on Rework
Build/Test Undiscovered Rework
<Initial Build/Test Work to Do>
Build/Test Effect of Design Undiscovered Rework On Fraction Correct
Build/Test Rework Discovery
Build/Test Work Believed to Be Done
<Initial Build/Test Work to Do>
Build/Test Delay in Discovering Rework
<Fraction Build/Test Rework Discovered>
Build/Test Fraction Reported Complete
Sensitivity of Build/Test Fraction Correct to Design Undiscovered Rework
Maximum Effect of Undiscovered Rework on Fraction Correct
Build/Test Effect of Build Undiscovered Rework on Fraction Correct
Build/Test Fraction of Work Believed Done Correct and Complete
<Design Effect of Build Work on Rework Discovery>
<Switch to Include Rework>
Sensitivity of Build/Test Fraction Correct to Build Undiscovered Rework
Maximum Sensitivity of Build/Test FCC to Design Undiscovered Rework
<Build/Test Rework to Do>
<Effect of Build Progress on Build/Test Rework Discovery>
Maximum Sensitivity of Build/Test Fraction Correct to Build Undiscovered Rework

Four Stock Rework Cycle

Design Phase of Work (Build/Test Similar)



End Design Rampup
Start Design Rampup
<Staff On Review>
Project Staff
Design Staff Total
<Time>
Design Startup
Design Staff
Design Planned Staff
Fraction Design Complete to Start Build Rampup
Design Fraction of Effort to Rework
Design Staff on Rework
Design Staff on Original Work
Design Relative Effort Required for Rework
Design Priority to Original Work
Design Indicated Staff
Design Maximum Work Rate Based on Rework Tasks Available
Design Maximum Work Rate Based on Original Work Tasks Available
Design Productivity on Rework
Design Original Work Being Accomplishmed
<Design Productivity>
Design Rework Being Accomplishmed
Design Minimum Time to Finish a Task
Design Initial Work to Do
Design Original Work to Do
Design Original Work Done Correctly
Design Rework Generation on Original Work
Design Fraction Reported Complete
Design Work Done
Design Rework Done Correctly
<Design Initial Work to Do>
Design Rework to Do
Normal Design Fraction Correct and Complete
Design Fraction Correct and Complete
Design Rework Generation on Rework
Design Undiscovered Rework
Design Rework Discovery
<Design Delay in Discovering Rework>
<Design Rework Discovered By Design>
<Design Remaining Rework Discovered by Build>
<Design Rework Discovered By
Design Work Believed to Be Done
Design Fraction of Work Done Correct and Complete
Design Initial Work to Do>
Maximum Effect of Undiscovered Rework on Fraction Correct>
Design Effect of Undiscovered Rework on Fraction Correct
Sensitivity of Design Fraction Correct to Design Undiscovered Rework
Maximum Sensitivity of Design Fraction Correct to Design Undiscovered Rework
<Fraction Rework Being Discovered>

Design Staff

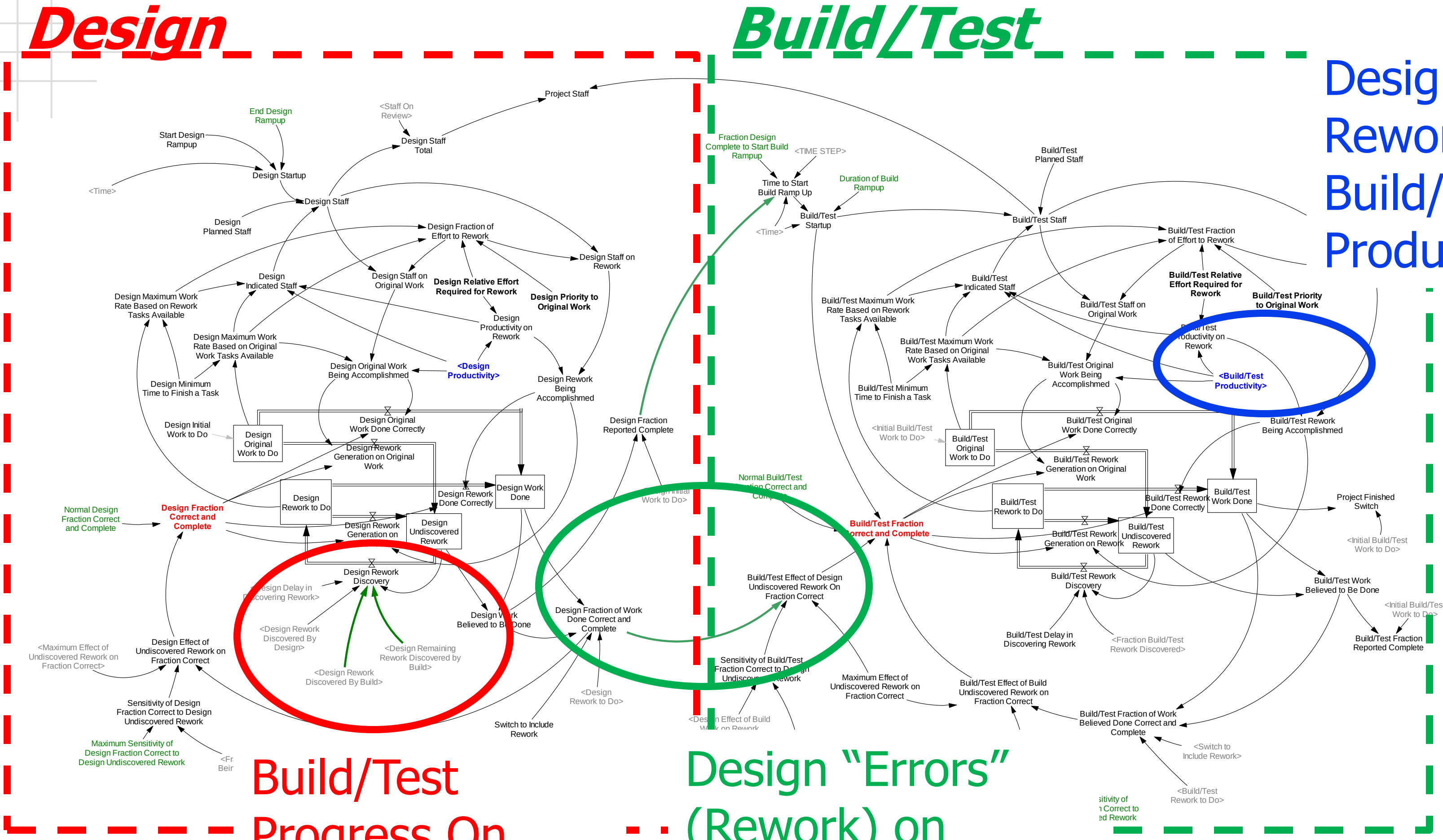
Allocation of Staff Based on Backlogs, etc.

"Errors Create Errors' Feedback

Rework Discovery

Massachusetts I

Phase Interconnections


Design
Build/Test
End Design Rampup
<Staff On Review>
Project Staff
Start Design Rampup
Design Staff Total
Design Startup
<Time>
Design Staff
Design Planned Staff
Design Fraction of Effort to Rework
Design Staff on Rework
Design Indicated Staff
Design Staff on Original Work
Design Relative Effort Required for Rework
Design Priority to Original Work
Design Maximum Work Rate Based on Rework Tasks Available
Design Productivity on Rework
Design Maximum Work Rate Based on Original Work Tasks Available
Design Original Work Being Accomplished
<Design Productivity>
Design Rework Being Accomplished
Design Minimum Time to Finish a Task
Design Initial Work to Do
Design Original Work to Do
Design Original Work Done Correctly
Design Rework Generation on Original Work
Design Fraction Reported Complete
Design Rework to Do
Design Rework Done Correctly
Design Work Done
Normal Design Fraction Correct and Complete
Design Fraction Correct and Complete
Design Rework Generation on
Design Undiscovered Rework
Design Rework Discovery
<Design Delay in Discovering Rework>
<Design Rework Discovered By Design>
<Design Remaining Rework Discovered by Build>
<Design Rework Discovered By Build>
Design Work Believed to Be Done
Design Fraction of Work Done Correct and Complete
<Maximum Effect of Undiscovered Rework on Fraction Correct>
Design Effect of Undiscovered Rework on Fraction Correct
<Design Rework to Do>
Sensitivity of Design Fraction Correct to Design Undiscovered Rework
Switch to Include Rework
Maximum Sensitivity of Design Fraction Correct to Design Undiscovered Rework
<Fr Beir
Fraction Design Complete to Start Build Rampup
<TIME STEP>
Time to Start Build Ramp Up
Duration of Build Rampup
Build/Test Startup
<Time>
Build/Test Planned Staff
Build/Test Staff
Build/Test Fraction of Effort to Rework
Build/Test Indicated Staff
Build/Test Relative Effort Required for Rework
Build/Test Priority to Original Work
Build/Test Maximum Work Rate Based on Rework Tasks Available
Build/Test Staff on Original Work
Build/Test Productivity on Rework
Build/Test Maximum Work Rate Based on Original Work Tasks Available
Build/Test Original Work Being Accomplished
<Build/Test Productivity>
Build/Test Minimum Time to Finish a Task
<Initial Build/Test Work to Do>
Build/Test Original Work to Do
Build/Test Original Work Done Correctly
Build/Test Rework Being Accomplished
Build/Test Rework Generation on Original Work
Normal Build/Test Fraction Correct and Complete
Build/Test Rework to Do
Build/Test Rework Done Correctly
Build/Test Work Done
Project Finished Switch
Build/Test Fraction Correct and Complete
Build/Test Rework Generation on Rework
Build/Test Undiscovered Rework
<Initial Build/Test Work to Do>
Build/Test Effect of Design Undiscovered Rework On Fraction Correct
Build/Test Rework Discovery
Build/Test Work Believed to Be Done
<Initial Build/Test Work to Do>
Build/Test Delay in Discovering Rework
<Fraction Build/Test Rework Discovered>
Build/Test Fraction Reported Complete
Sensitivity of Build/Test Fraction Correct to Design Undiscovered Rework
Maximum Effect of Undiscovered Rework on Fraction Correct
Build/Test Effect of Build Undiscovered Rework on Fraction Correct
<Design Effect of Build Work on Rework
Build/Test Fraction of Work Believed Done Correct and Complete
<Switch to Include Rework>
<Build/Test Rework to Do>
Sensitivity of Fraction Correct to Undiscovered Rework

Design Rework on Build/Test Productivity

Build/Test Progress On Design Rework Discovery

Design "Errors" (Rework) on Build/Test Fraction Correct

Massachusetts Institute of Technology