ESD.36 System Project Management

Lecture 8

Project Dynamics: Feedbacks

Instructor(s)

Dr. James Lyneis

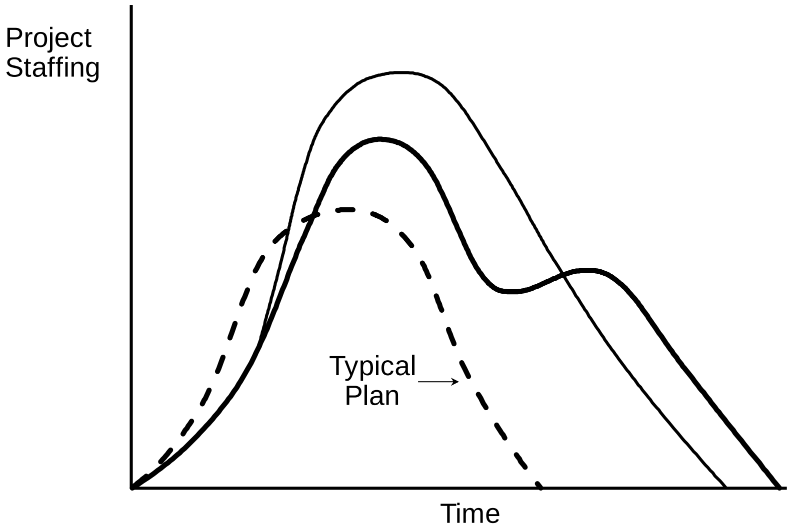
October 2, 2012

Massachusetts Institute of Technology

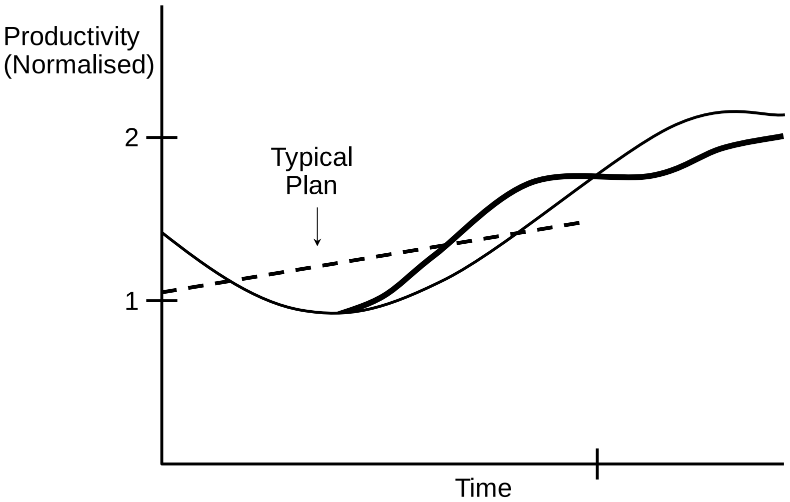
Today's Agenda

- Project Dynamics: Feedback Loops
- Qualitative Lessons
- Quantitative Models
- Validation and Model Extensions (if time)

The Problem …



Project
Staffing
Typical
Plan
Time

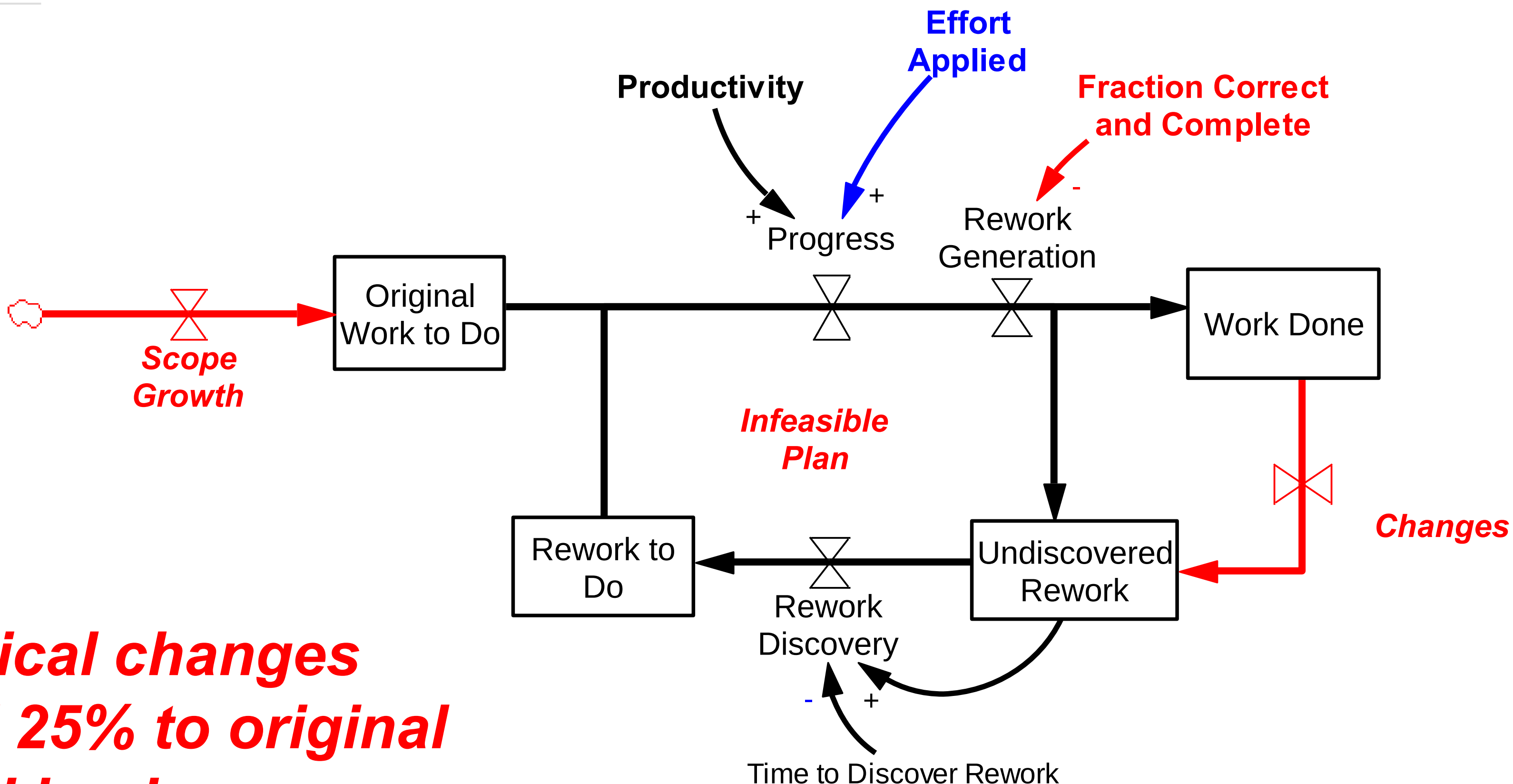


Productivity
(Normalised)
2
1
Typical
Plan
Time

Dynamics of Project Performance

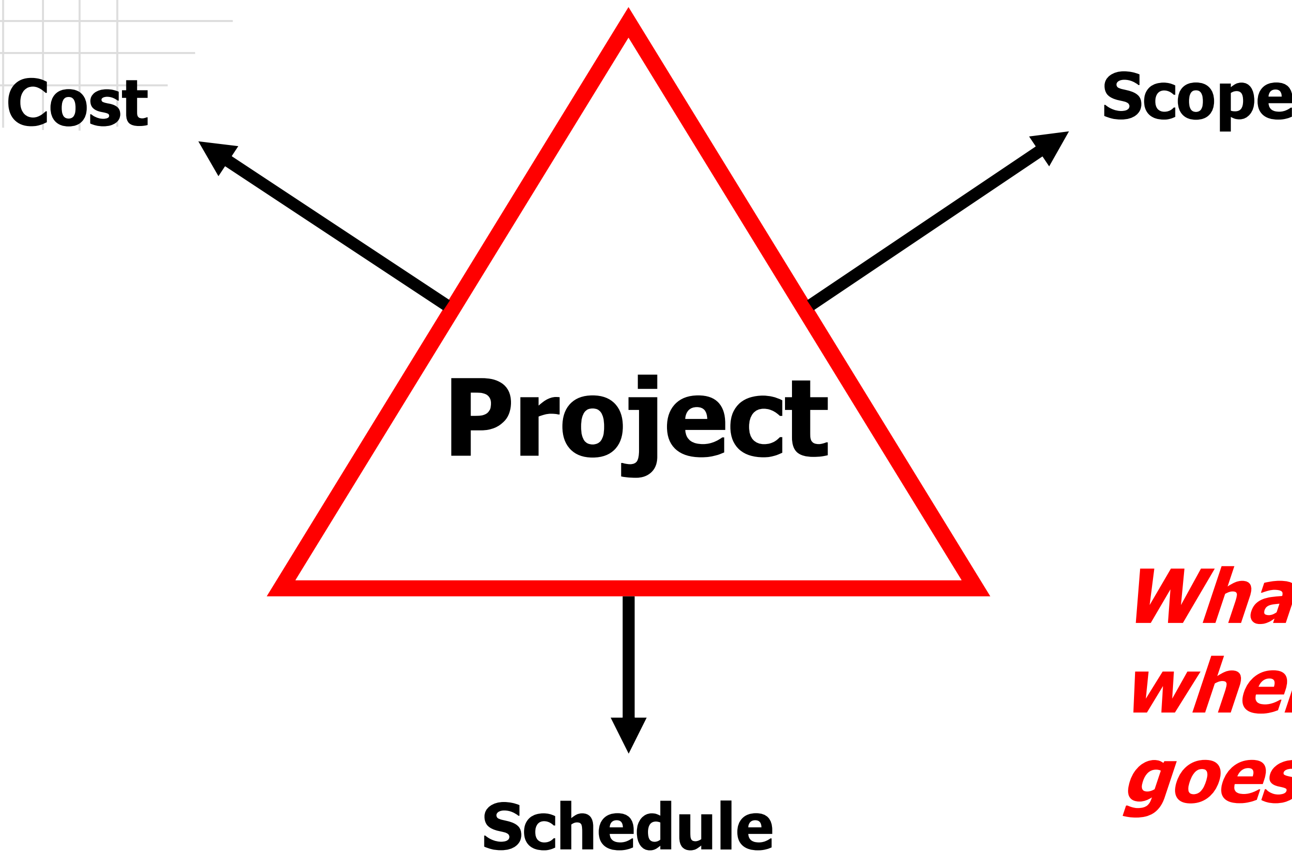
- The “rework cycle”
  - Fraction correct and complete
  - Undiscovered rework
- **Feedback effects on productivity and fraction correct**
  - **Negative, controlling**
  - **Positive, re-enforcing, often “vicious circles”**
- Knock-on effects between work phases
  - Availability and quality of work products
  - Progress to discover upstream rework

Something Goes Wrong



Typical changes add 25% to original workload

The "Iron Triangle"



What happens when something goes wrong?

When your project falls behind schedule, what can you do to get it back on track?

1. Add people?
2. Work longer hours?
3. Work more "intensely" (including cutting corners, increasing concurrency, releasing work earlier than ideal)?
4. Slip the schedule?
5. Cut scope?
6. Other?

We'll discuss results in my lecture in two weeks.

What do you do at ~30% complete?

What is your (company's) response? Put a 1 next to your primary response, at 2 next to your secondary response, and so on. If you would not use a response, leave it blank, otherwise try to rank the options even if you rarely use them in practice.

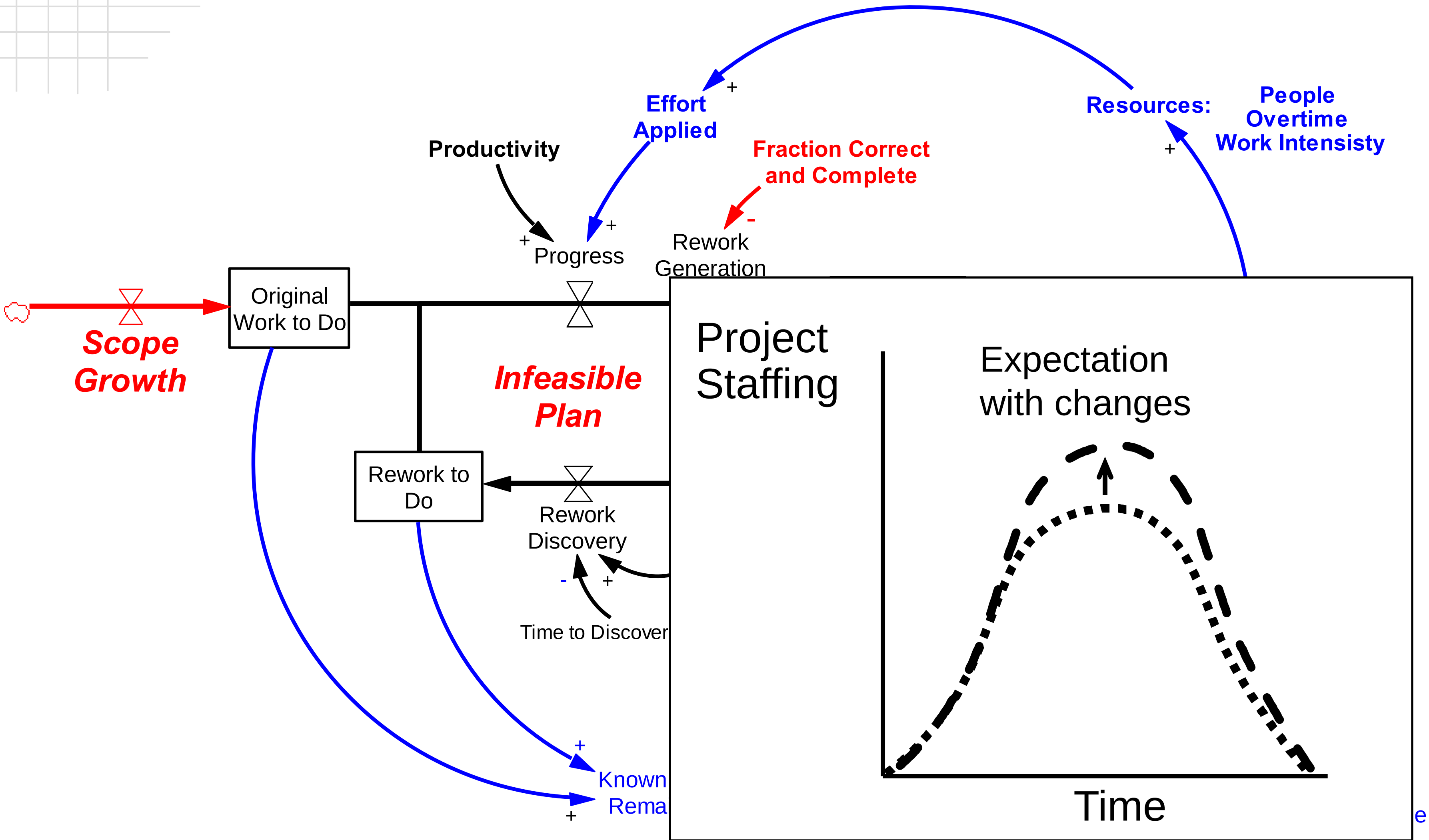
1. Add people?
2. Work longer hours?
3. Work more "intensely" (including cutting corners, increasing concurrency, releasing work earlier than ideal)?
4. Slip the schedule?
5. Cut scope?
6. Other?

What do you do at ~65% complete?

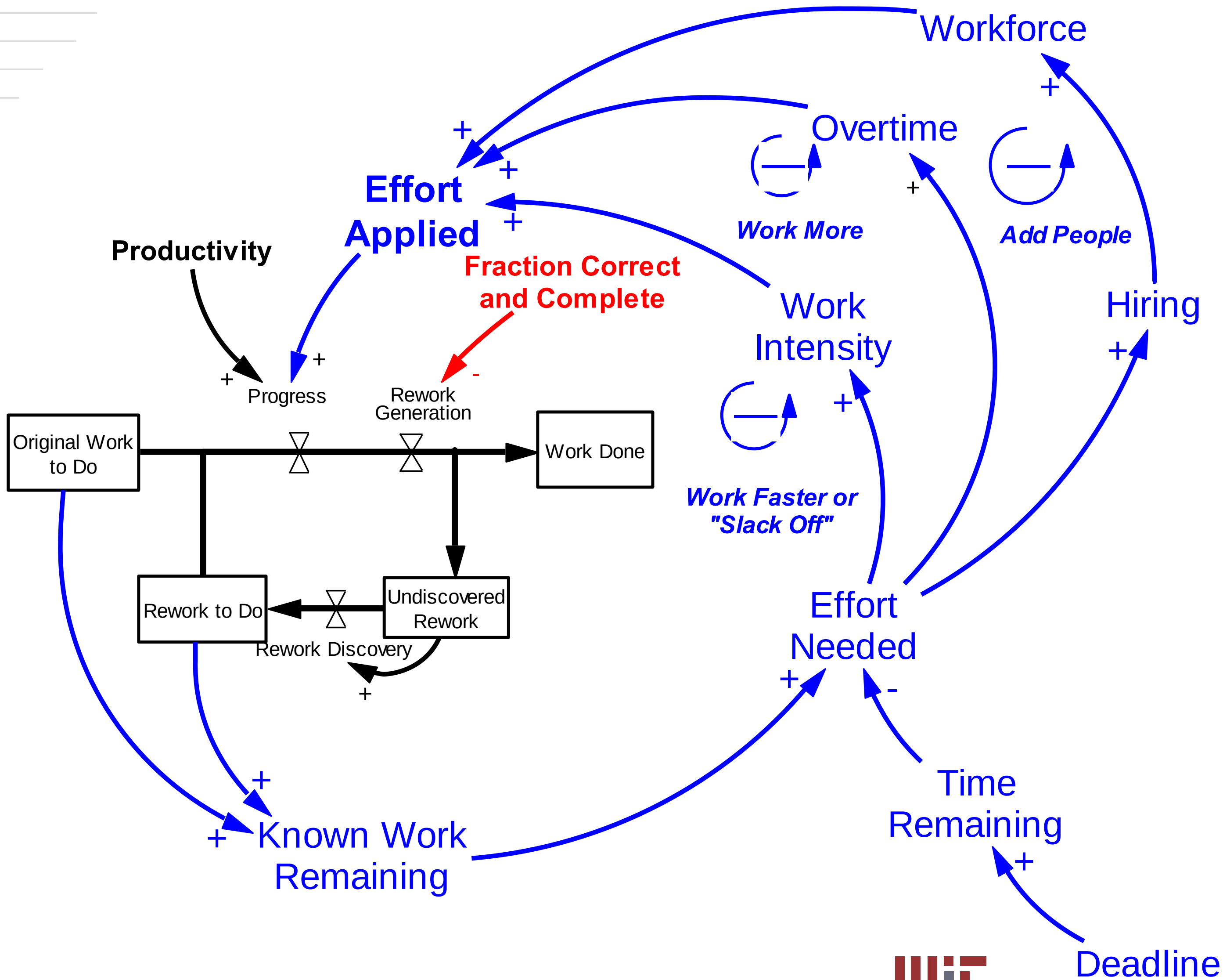
What is your (company's) response? Put a 1 next to your primary response, at 2 next to your secondary response, and so on. If you would not use a response, leave it blank, otherwise try to rank the options even if you rarely use them in practice.

1. Add people?
2. Work longer hours?
3. Work more "intensely" (including cutting corners, increasing concurrency, releasing work earlier than ideal)?
4. Slip the schedule?
5. Cut scope?
6. Other?

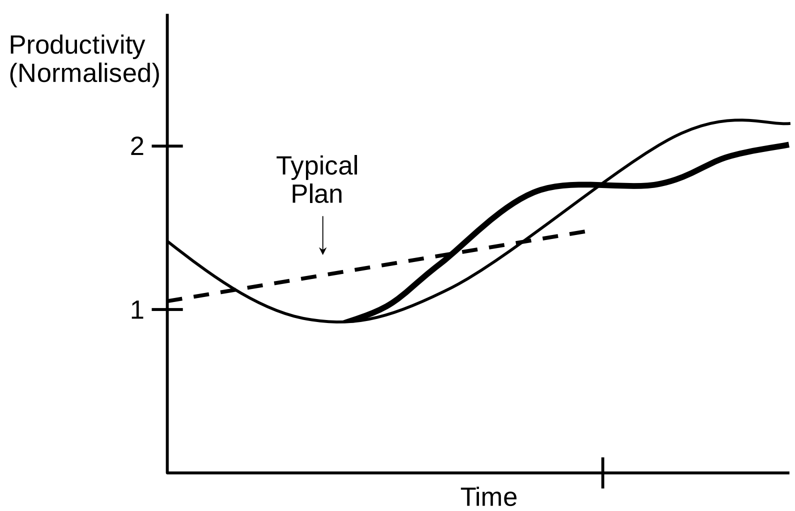
Management Reacts …

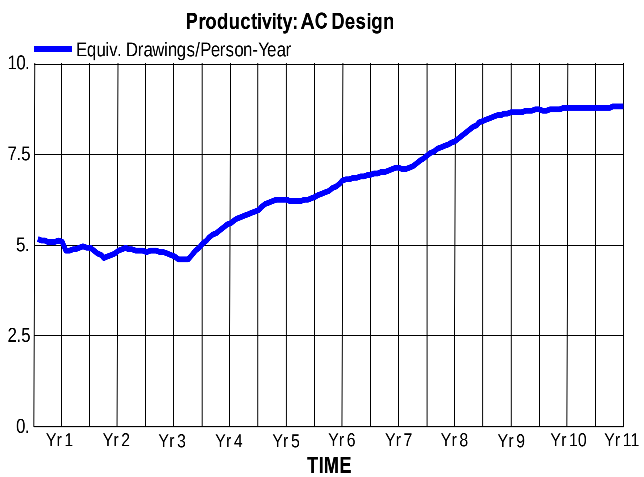


Management Reacts …



On a Typical Project, Productivity & Fraction Correct Vary Over Time



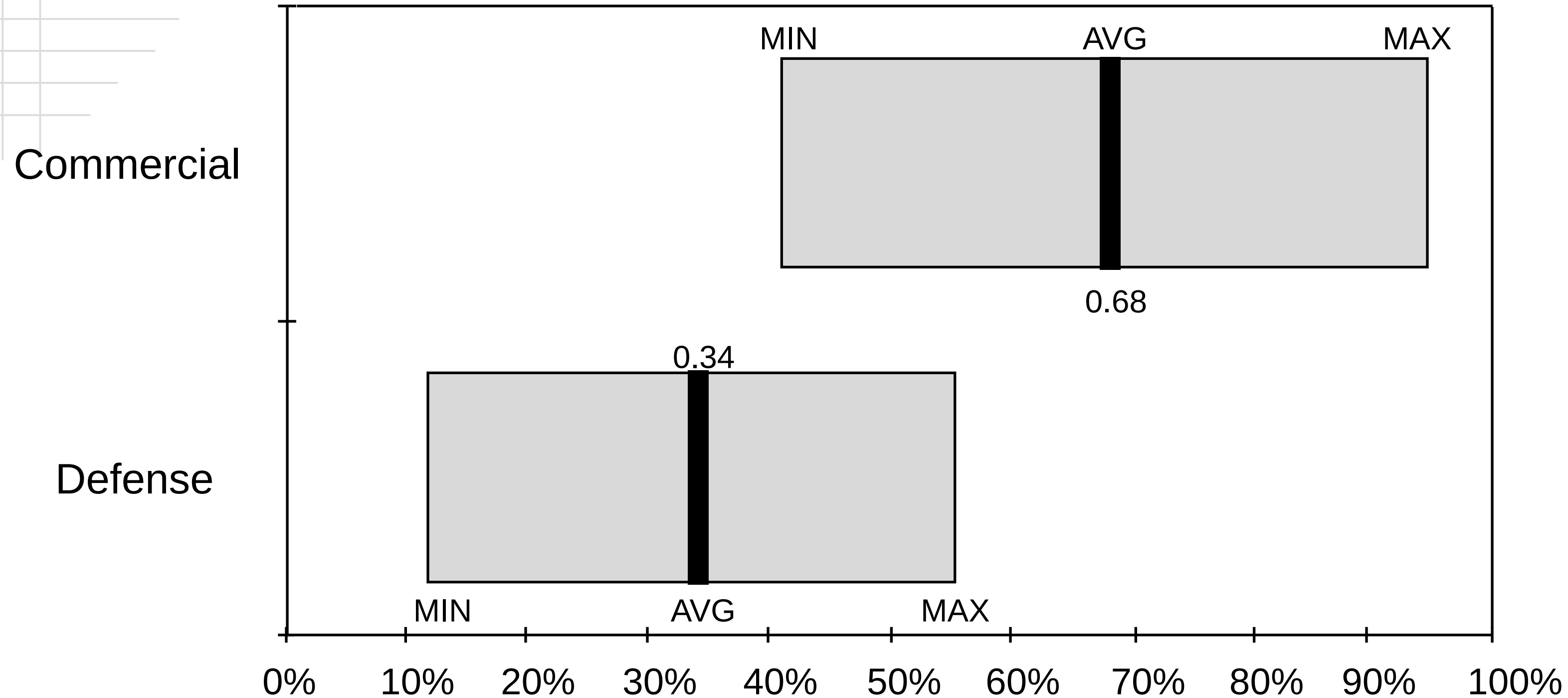


Why?

Why?

- ***Side-effect feedbacks (often "vicious circles")***
- ***Knock-on or domino effects within or between work phases***
- *Knock-on or domino effects between projects*

Benchmarking Data -- Average 'Quality'

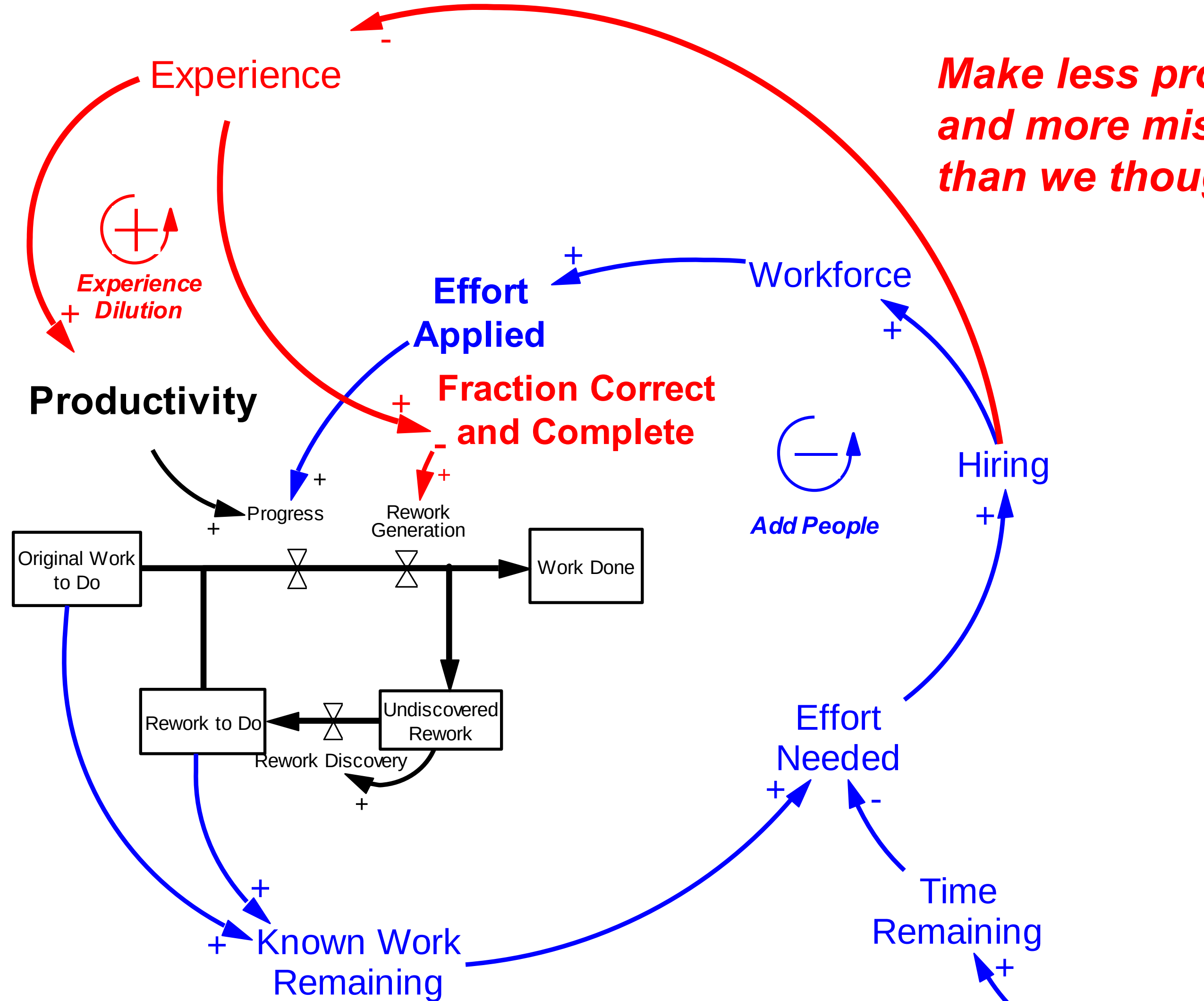


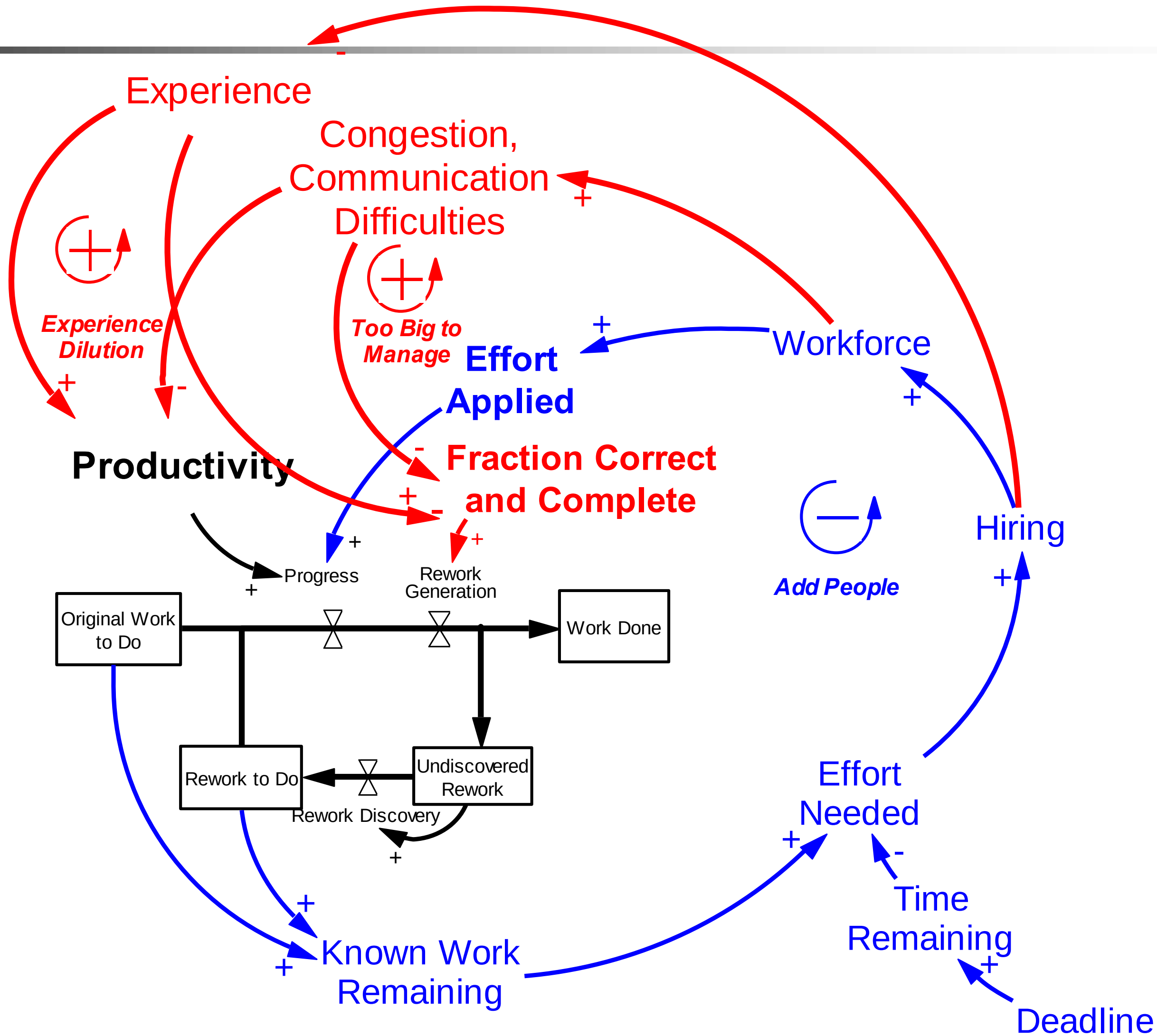
[From a survey of 21 software development projects (14 commercial, 7 defense, 6 companies); Cooper, K.G. and T. W. Mullen. 1993. Swords and Plowshares: The Rework Cycles of Defense and Commercial Software Development Projects *American Programmer*, May edition.] B

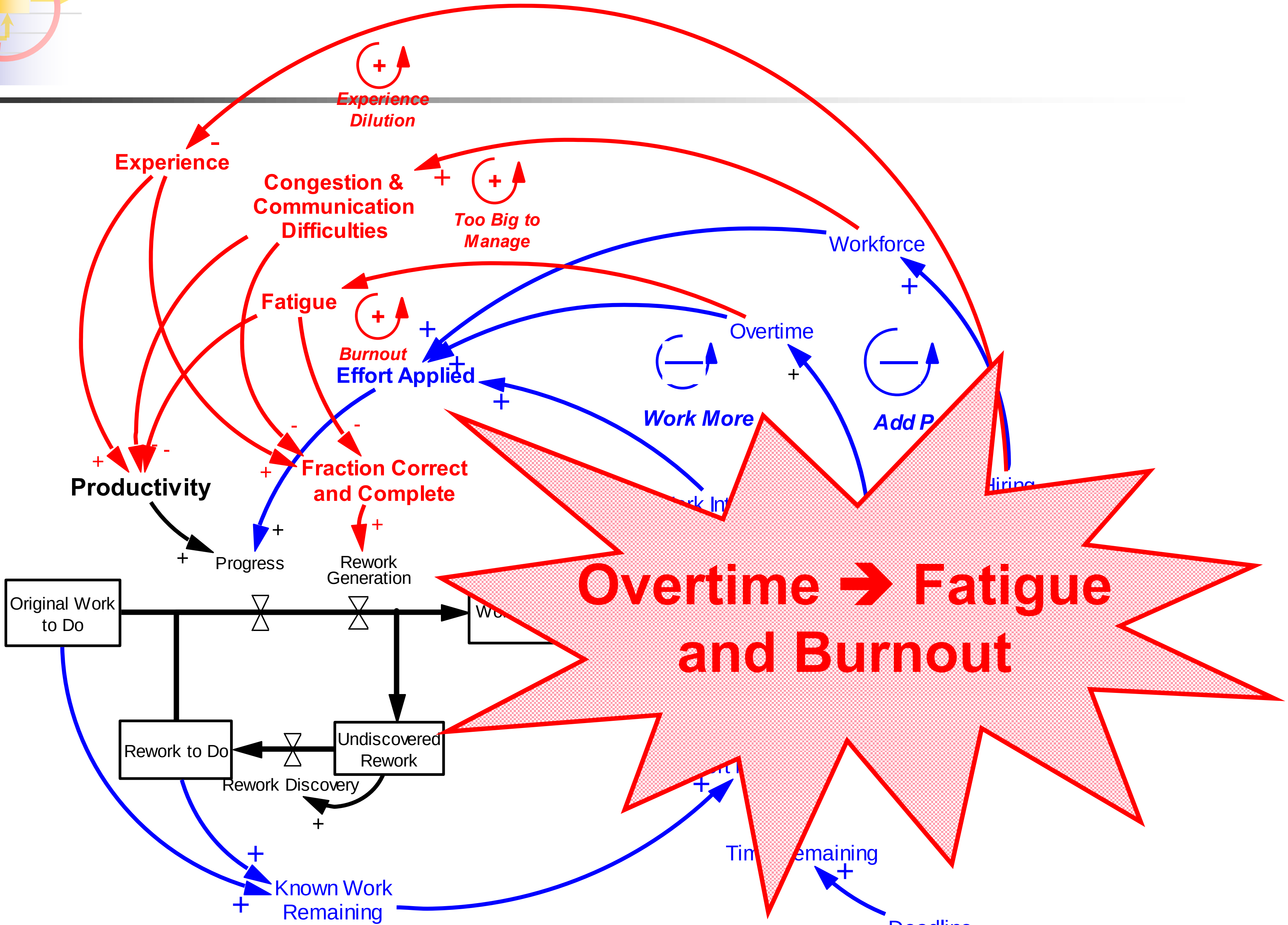
Actions and Consequences

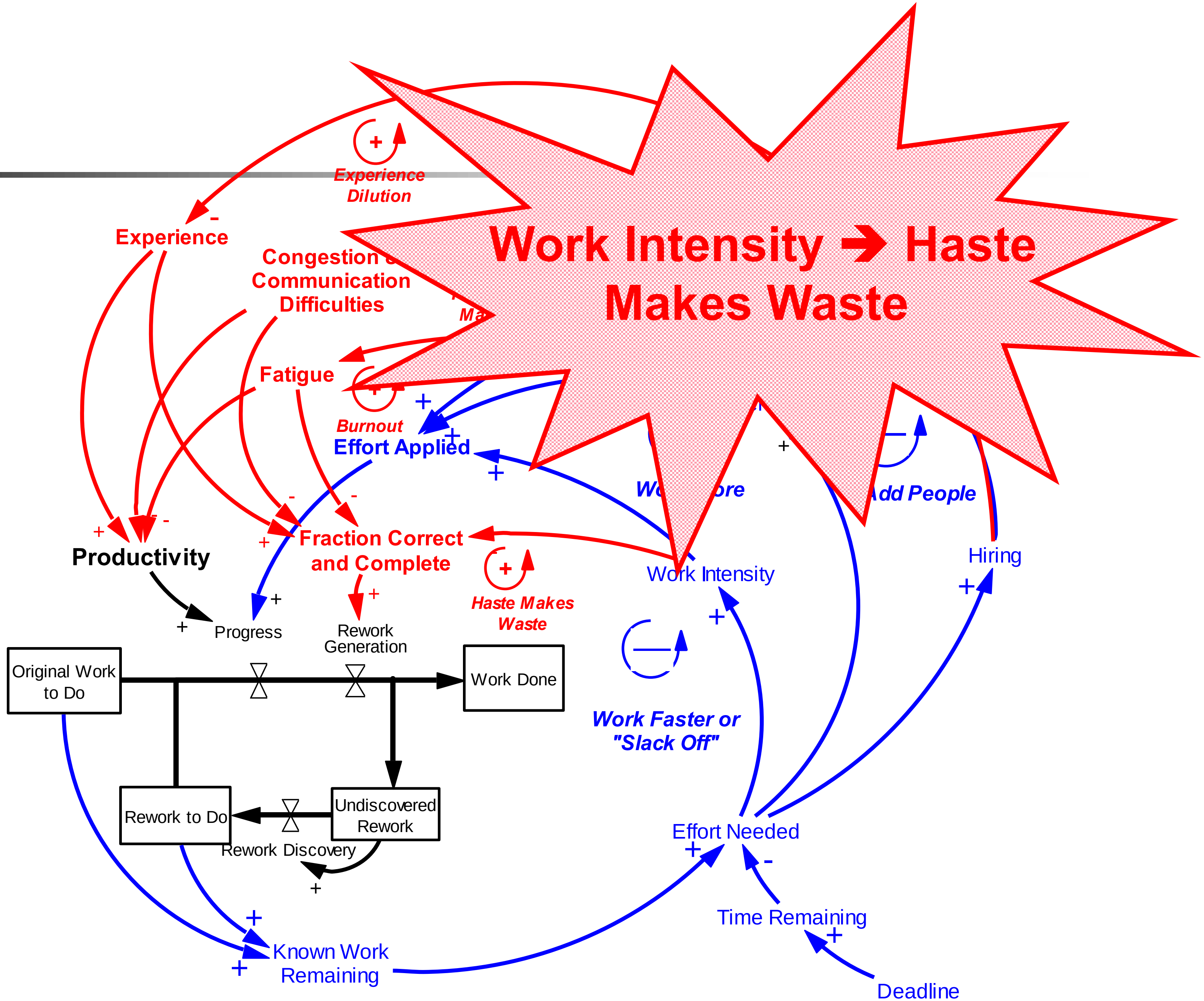
Control Action	Side Effects
Hiring	
Overtime	
Work more intensely	

Each Controlling Action Initiates Vicious Circles…

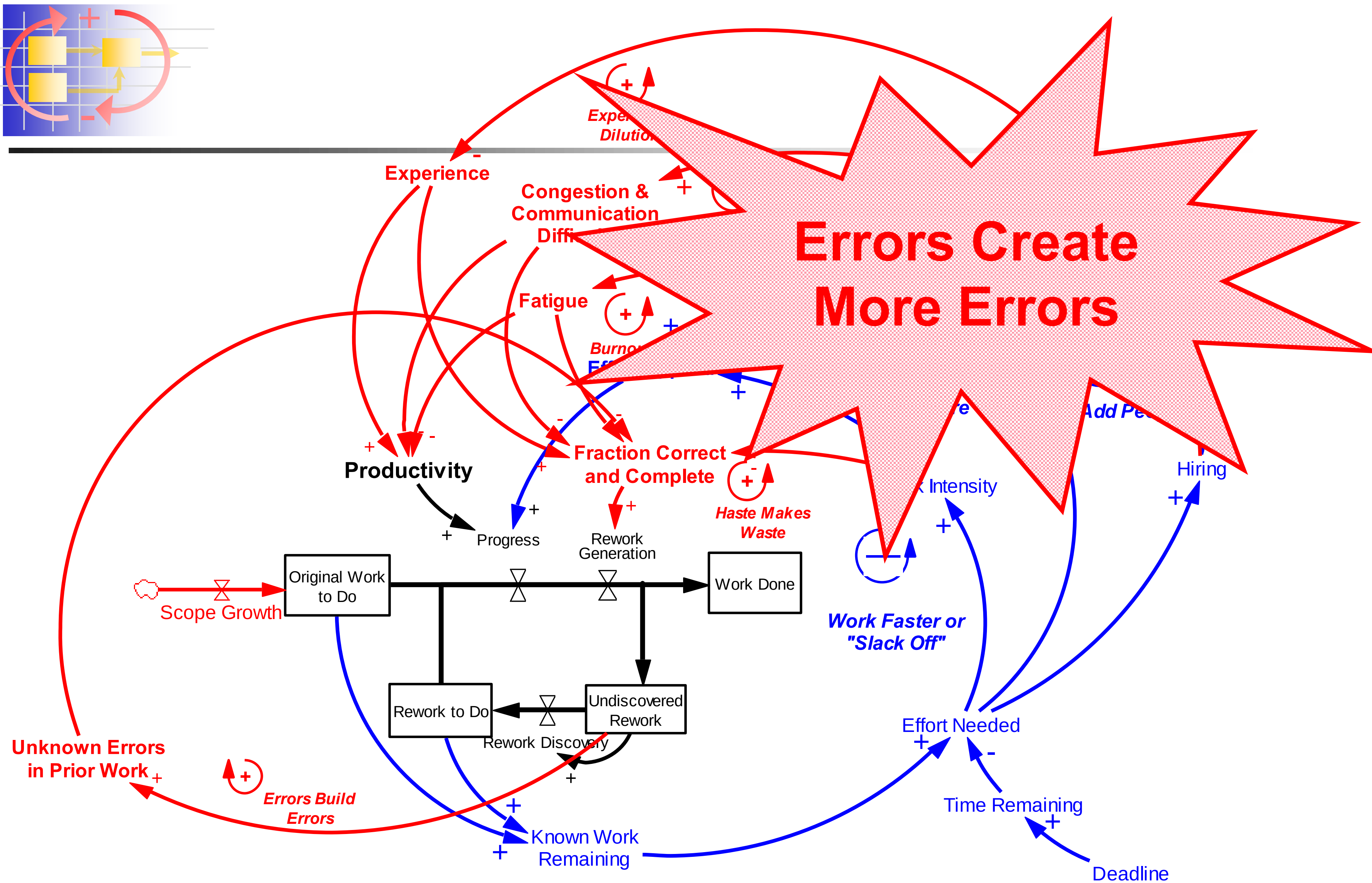



Experience
Congestion,
Communication
Difficulties
Experience
Dilution
Too Big to
Manage
Effort
Applied
Workforce
Productivity
Fraction Correct
and Complete
Hiring
Add People
Progress
Rework
Generation
Original Work
to Do
Work Done
Rework to Do
Undiscovered
Rework
Rework Discovery
Effort
Needed
Time
Remaining
Known Work
Remaining
Deadline

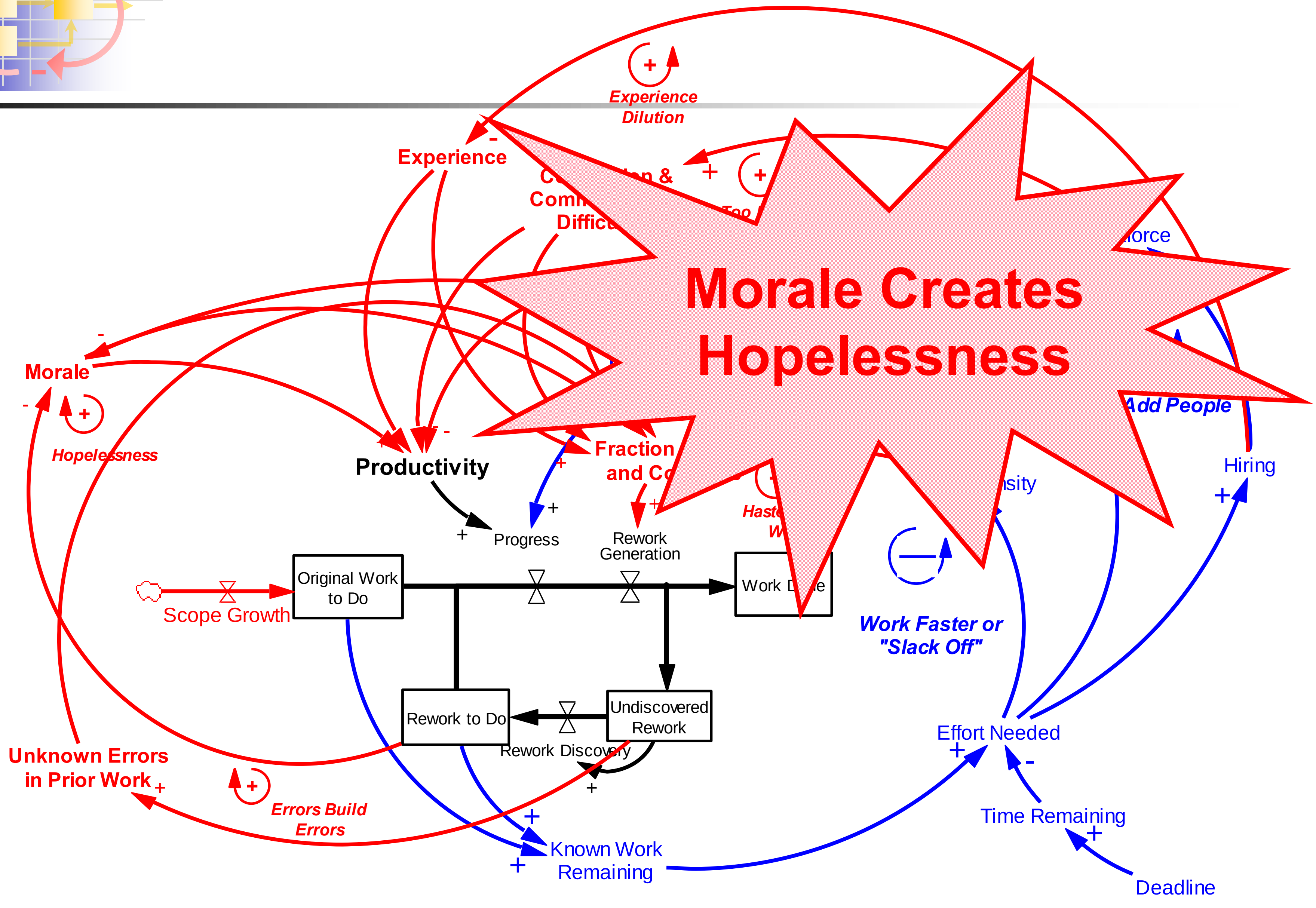

Experience Dilution
Experience
Congestion & Communication Difficulties
Too Big to Manage
Workforce
Fatigue
Burnout
Effort Applied
Overtime
Work More
Add P
Productivity
Fraction Correct and Complete
Progress
Rework Generation
Original Work to Do
Rework to Do
Undiscovered Rework
Rework Discovery
Known Work Remaining
Deadline
Overtime → Fatigue and Burnout

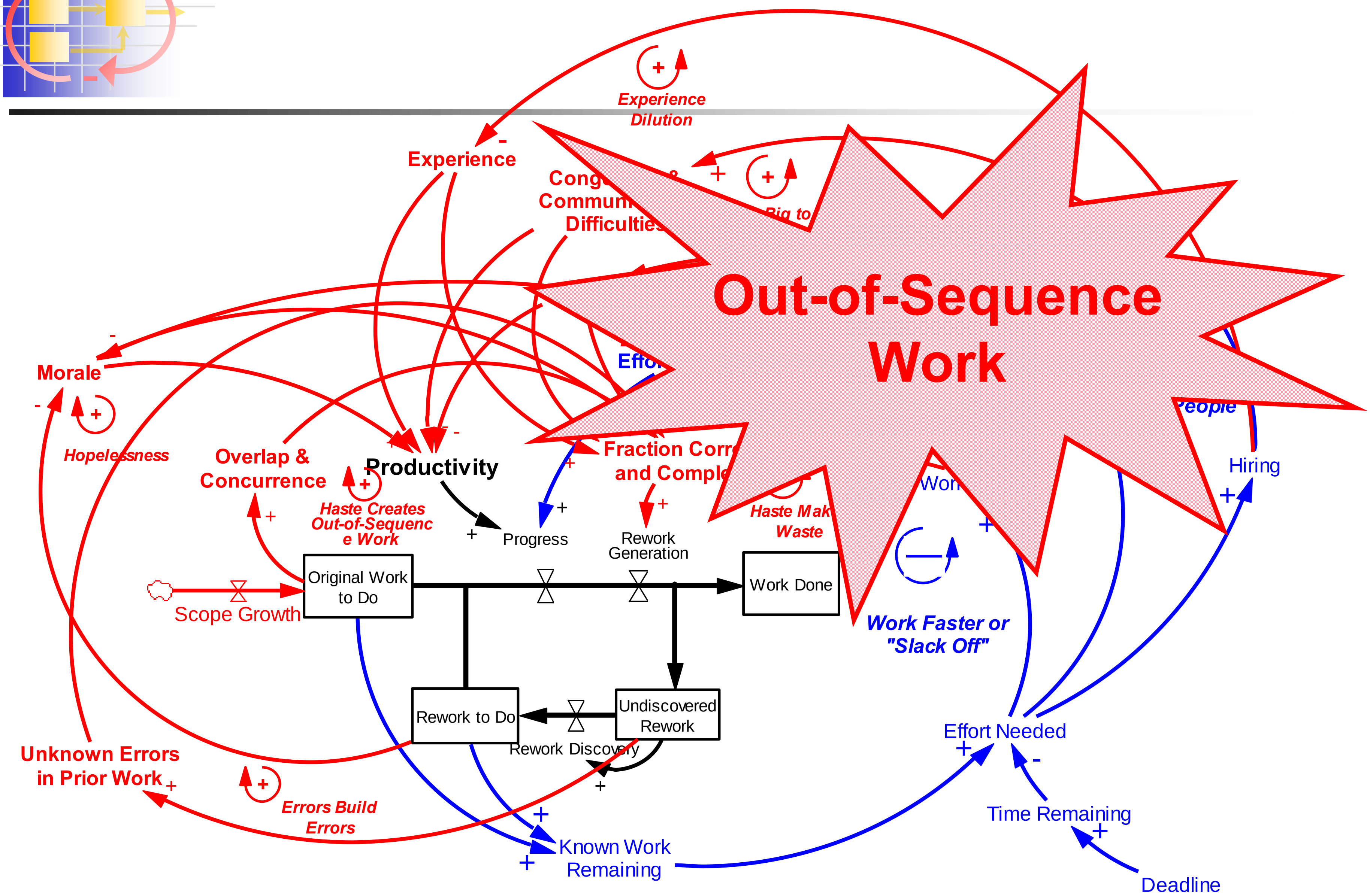


Work Intensity ➔ Haste Makes Waste
Experience Dilution
Experience
Congestion
Communication Difficulties
Fatigue
Burnout
Effort Applied
Productivity
Fraction Correct and Complete
Haste Makes Waste
Work Intensity
Add People
Hiring
Progress
Rework Generation
Original Work to Do
Work Done
Work Faster or "Slack Off"
Rework to Do
Undiscovered Rework
Rework Discovery
Effort Needed
Time Remaining
Known Work Remaining
Deadline

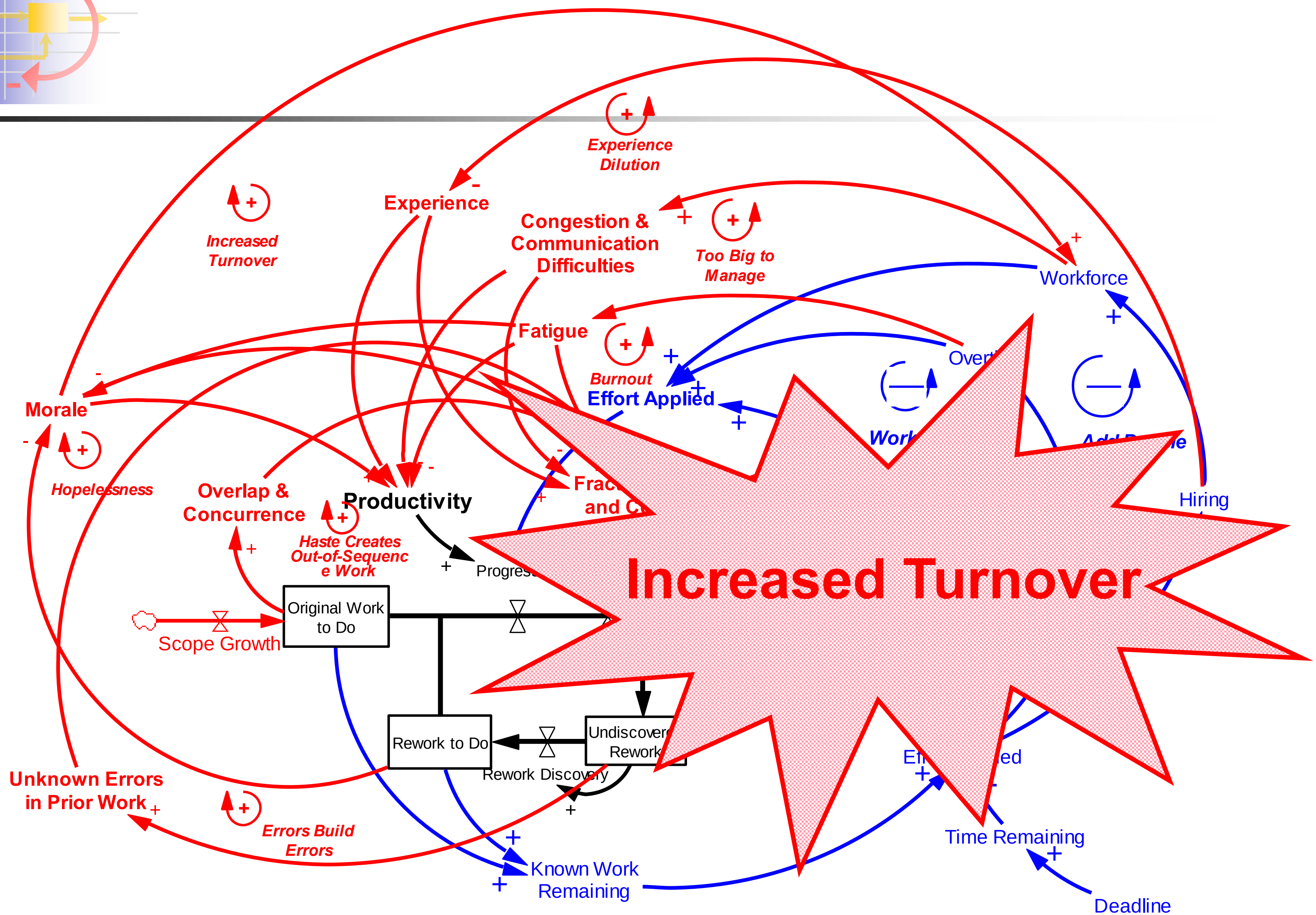


Errors Create More Errors
Experience
Congestion & Communication
Fatigue
Productivity
Fraction Correct and Complete
Haste Makes Waste
Progress
Rework Generation
Original Work to Do
Work Done
Scope Growth
Rework to Do
Undiscovered Rework
Rework Discovery
Unknown Errors in Prior Work
Errors Build Errors
Known Work Remaining
Work Faster or "Slack Off"
Intensity
Hiring
Effort Needed
Time Remaining
Deadline

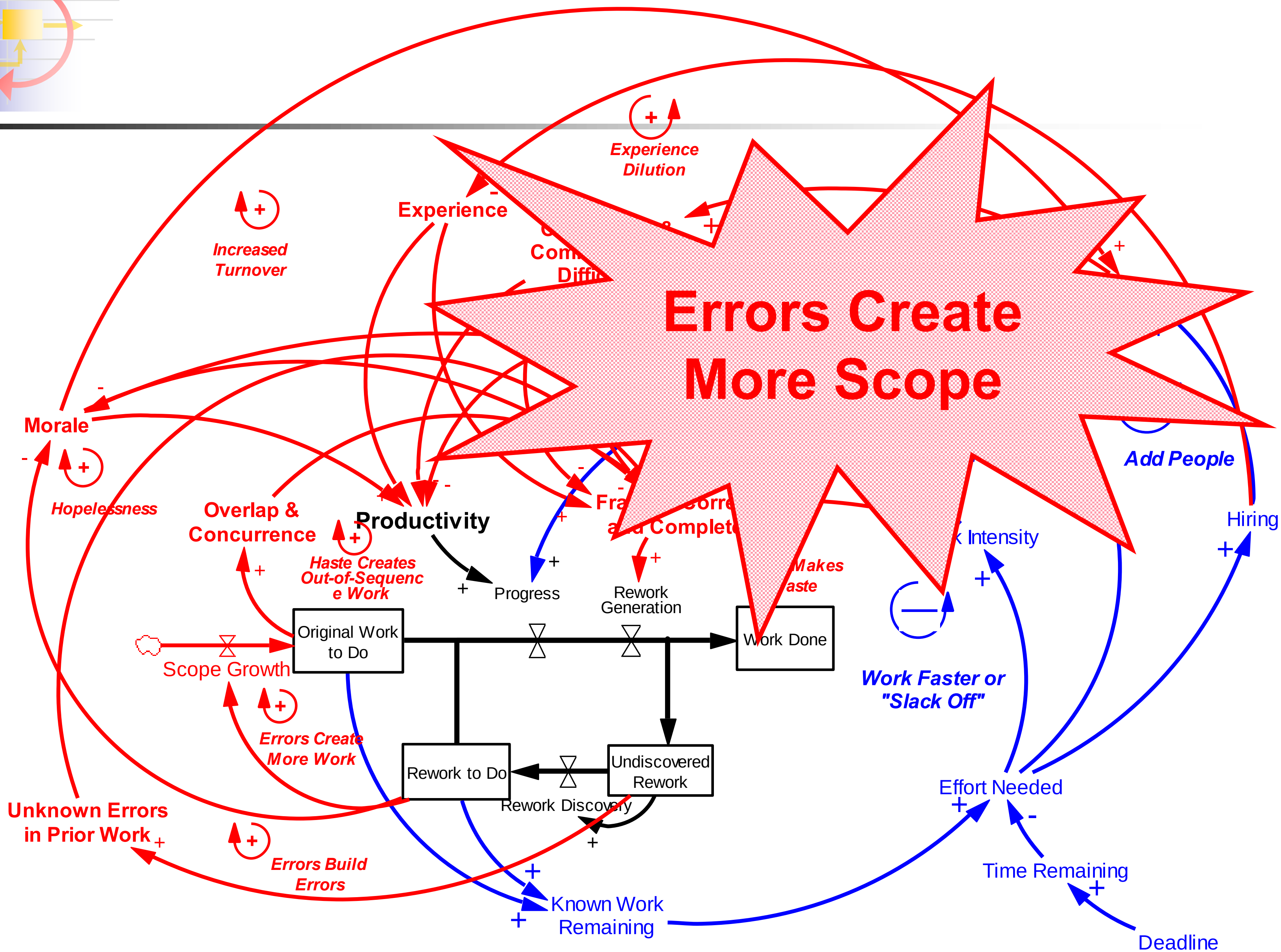

Morale Creates Hopelessness
Experience Dilution
Experience
Morale
Hopelessness
Productivity
Fraction
Progress
Rework Generation
Scope Growth
Original Work to Do
Work Done
Rework to Do
Undiscovered Rework
Rework Discovery
Unknown Errors in Prior Work
Errors Build Errors
Known Work Remaining
Work Faster or "Slack Off"
Effort Needed
Time Remaining
Deadline
Hiring
Add People

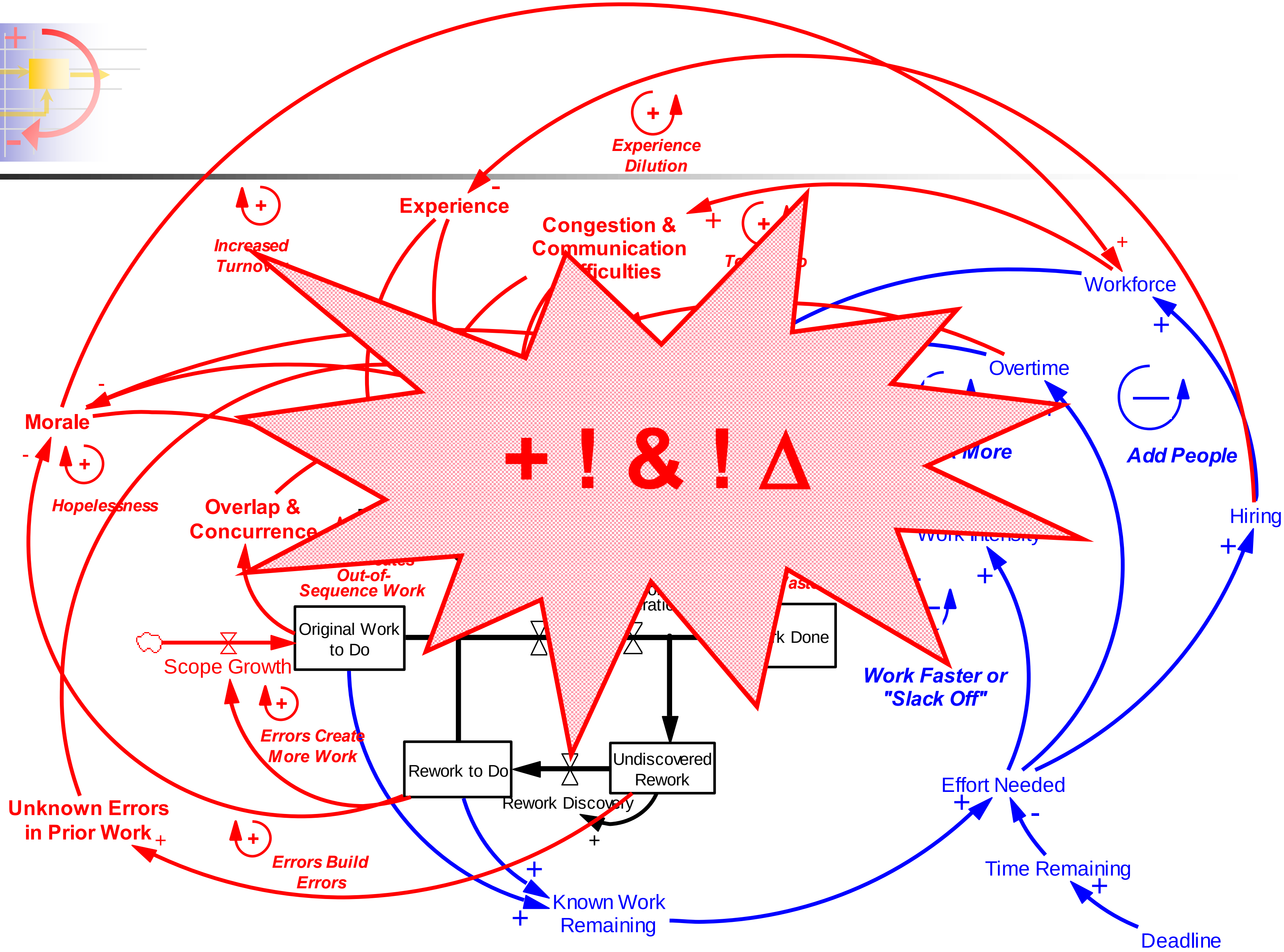


Experience Dilution
Experience
Congestion & Communication Difficulties
Too Big to ...
Out-of-Sequence Work
Morale
Hopelessness
Overlap & Concurrence
Productivity
Haste Creates Out-of-Sequence Work
Progress
Rework Generation
Fraction Correct and Complete
Haste Makes Waste
Effort
People
Hiring
Work Faster or "Slack Off"
Original Work to Do
Scope Growth
Work Done
Rework to Do
Undiscovered Rework
Rework Discovery
Unknown Errors in Prior Work
Errors Build Errors
Known Work Remaining
Effort Needed
Time Remaining
Deadline



Experience Dilution
Experience
Increased Turnover
Congestion & Communication Difficulties
Too Big to Manage
Workforce
Fatigue
Burnout
Effort Applied
Morale
Hopelessness
Overlap & Concurrence
Productivity
Haste Creates Out-of-Sequence Work
Original Work to Do
Scope Growth
Rework to Do
Undiscovered Rework
Rework Discovery
Unknown Errors in Prior Work
Errors Build Errors
Known Work Remaining
Time Remaining
Deadline
Hiring
Increased Turnover


Errors Create More Scope
Experience Dilution
Increased Turnover
Experience
Morale
Hopelessness
Overlap & Concurrence
Productivity
Haste Creates Out-of-Sequence Work
Progress
Rework Generation
Work Done
Original Work to Do
Scope Growth
Errors Create More Work
Rework to Do
Undiscovered Rework
Rework Discovery
Unknown Errors in Prior Work
Errors Build Errors
Known Work Remaining
Effort Needed
Time Remaining
Deadline
Work Faster or "Slack Off"
Add People
Hiring

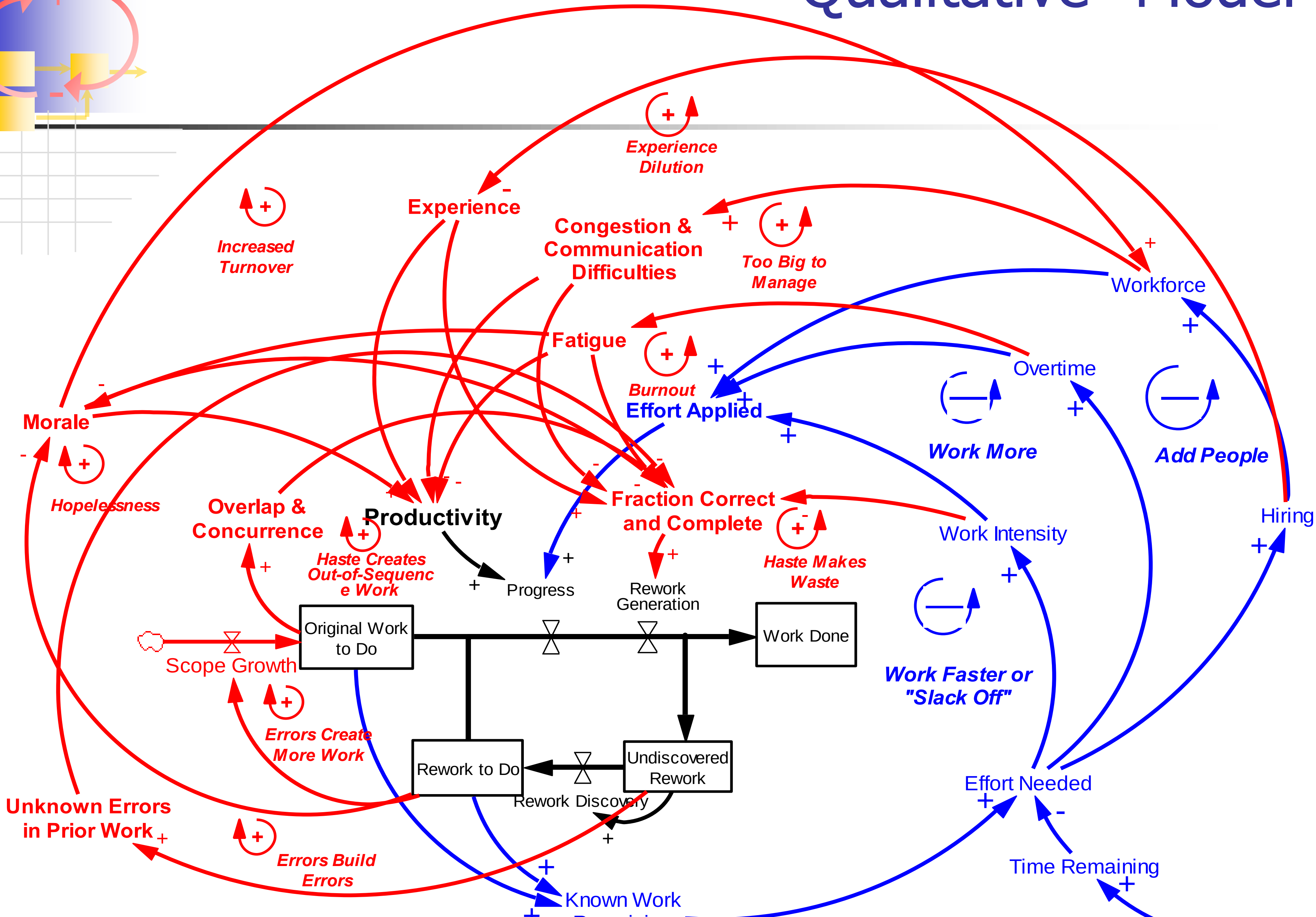


Experience Dilution
Experience
Increased Turnov
Congestion & Communication ...fficulties
+ ! & ! Δ
Workforce
Overtime
Add People
Hiring
Morale
Hopelessness
Overlap & Concurrence
Out-of-Sequence Work
Original Work to Do
Scope Growth
Errors Create More Work
Rework to Do
Undiscovered Rework
Rework Discovery
Work Faster or "Slack Off"
Effort Needed
Time Remaining
Deadline
Unknown Errors in Prior Work
Errors Build Errors
Known Work Remaining

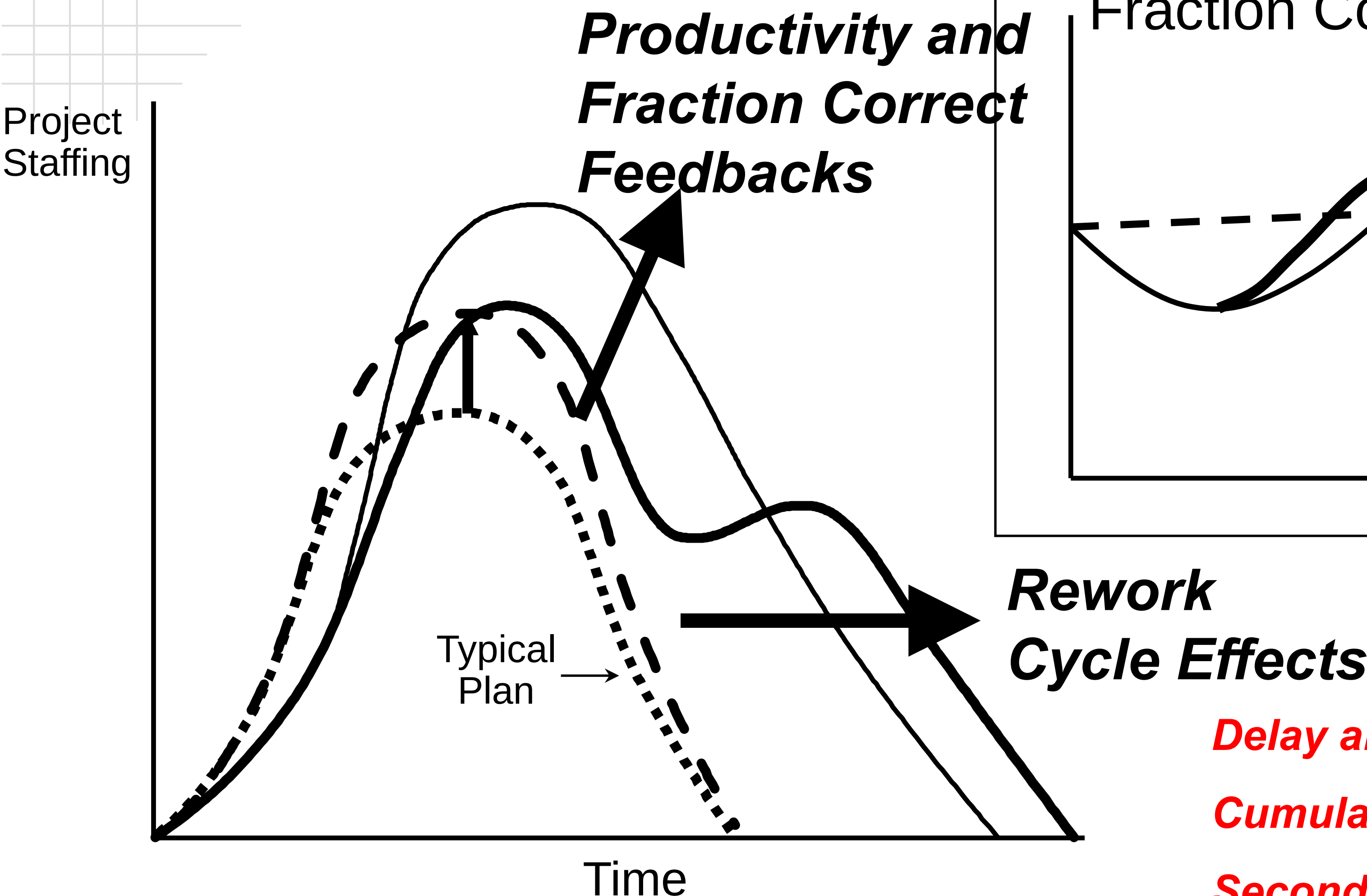
Today's Agenda

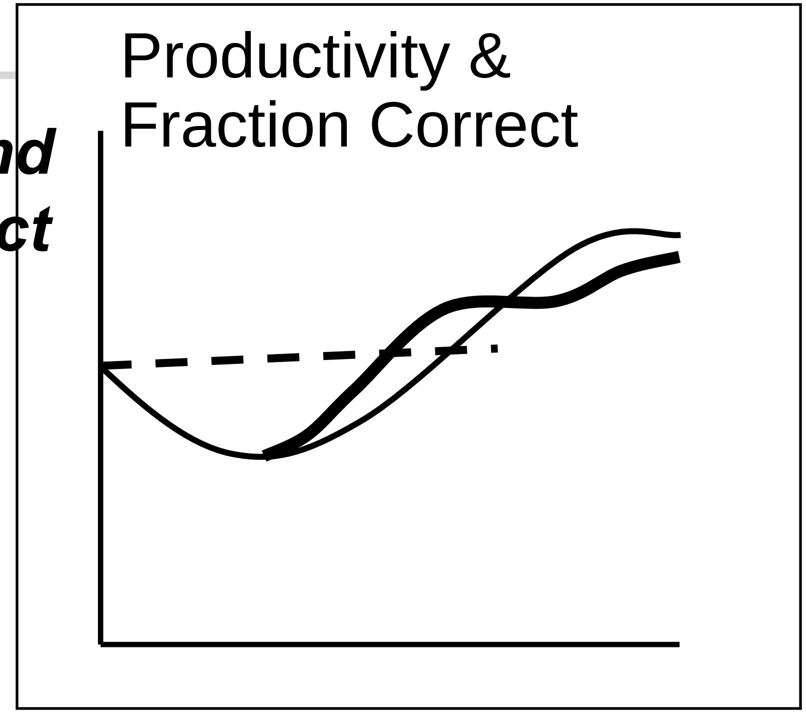
- Project Dynamics: Feedback Loops
- Qualitative Lessons
- Quantitative Models
- Validation and Model Extensions

"Qualitative" Model



Project Behavior





Delay and disruption

Cumulative Impact

Secondary impact

Ripple effects

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
- Management actions to control a project can initiate reinforcing feedback loops that cause project problems to “snowball” and amplify costs far in excess of the cost of triggering event
- “Soft Factors” such as fatigue and morale can be big drivers of productivity loss and rework

Survey Question

In your organization, what do you estimate is the relative contribution of the direct costs of External Changes and the costs of Management Responses to project overruns:

1. Costs of External Changes greater than costs of Management Responses
2. Costs of Management Responses greater than costs of External Changes
3. Costs of both about same
4. Varies too much by project to say for sure

Today's Agenda

- Project Dynamics: Feedback Loops
- Qualitative Lessons
- Quantitative Models
- Validation and Model Extensions

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

Focus – Development of Computer Models

- More detailed stock/flow – causal diagrams
- Details of policy controls and side effects
- Some equations
- More detail in textbook chapter SD3

Purpose

- Understand enough of how model works to
  - Understand simulation results in next lectures
  - Execute policy tests and explain results in HW#3 and HW#5
  - Generate insights into improved practice
- Use and extend model on projects or other applications

We will use two models …

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

Project Model 1: Simple Rework Cycle

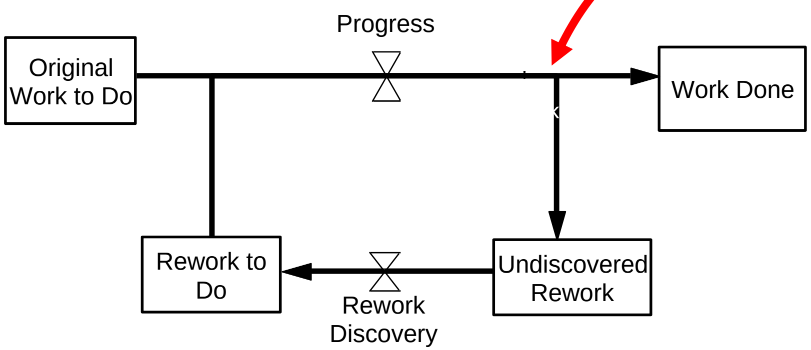
- Rework cycle model (HW#3)
  - Three stocks
  - Variable rework discovery time
  - “Errors Build Errors” Feedback

Model you develop in Part 1 of HW#3

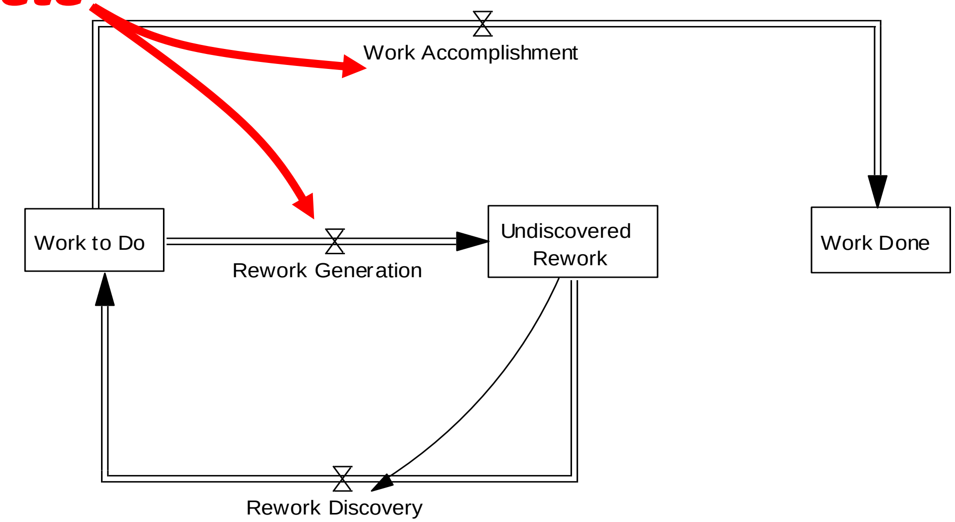
Two Views of the Rework Cycle

Fraction Correct & Complete

Complete Model

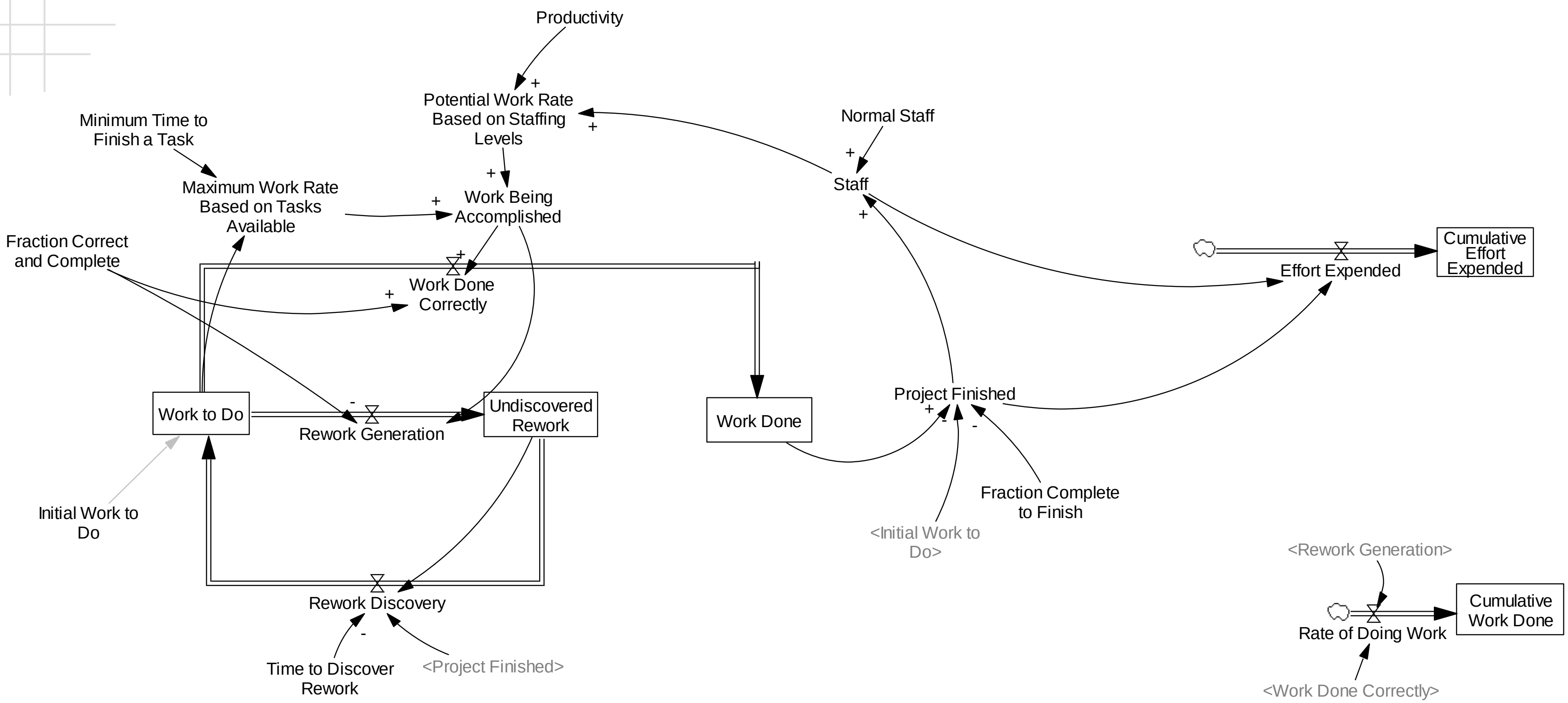


Simplified Version



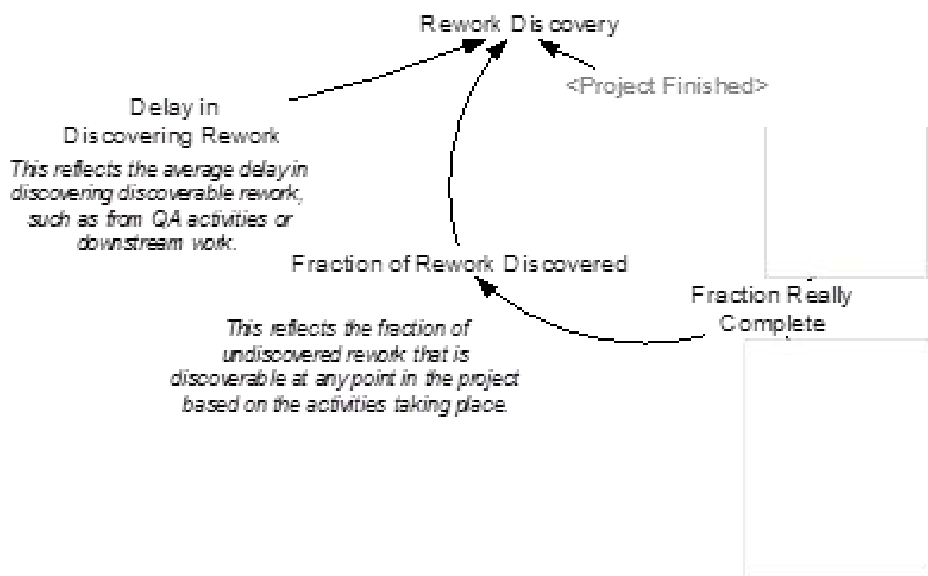
The simplified version assumes that rework tasks require the same effort as original tasks, and that it is not important to distinguish between original work and rework.

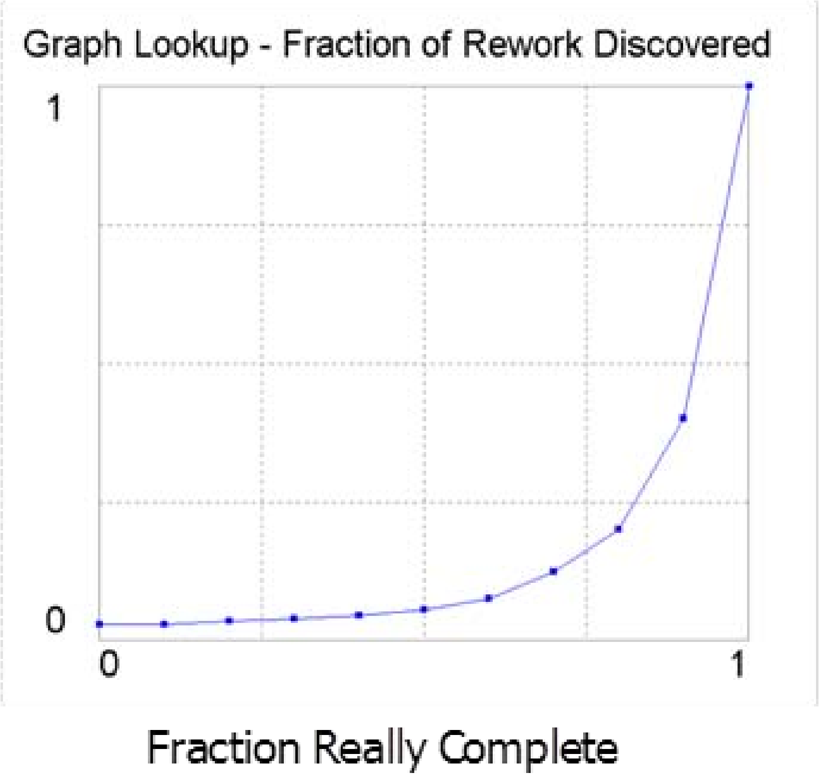
Simple Rework Cycle Model



Steps 1-4 of Homework #3

Rework Discovery Depends on Progress

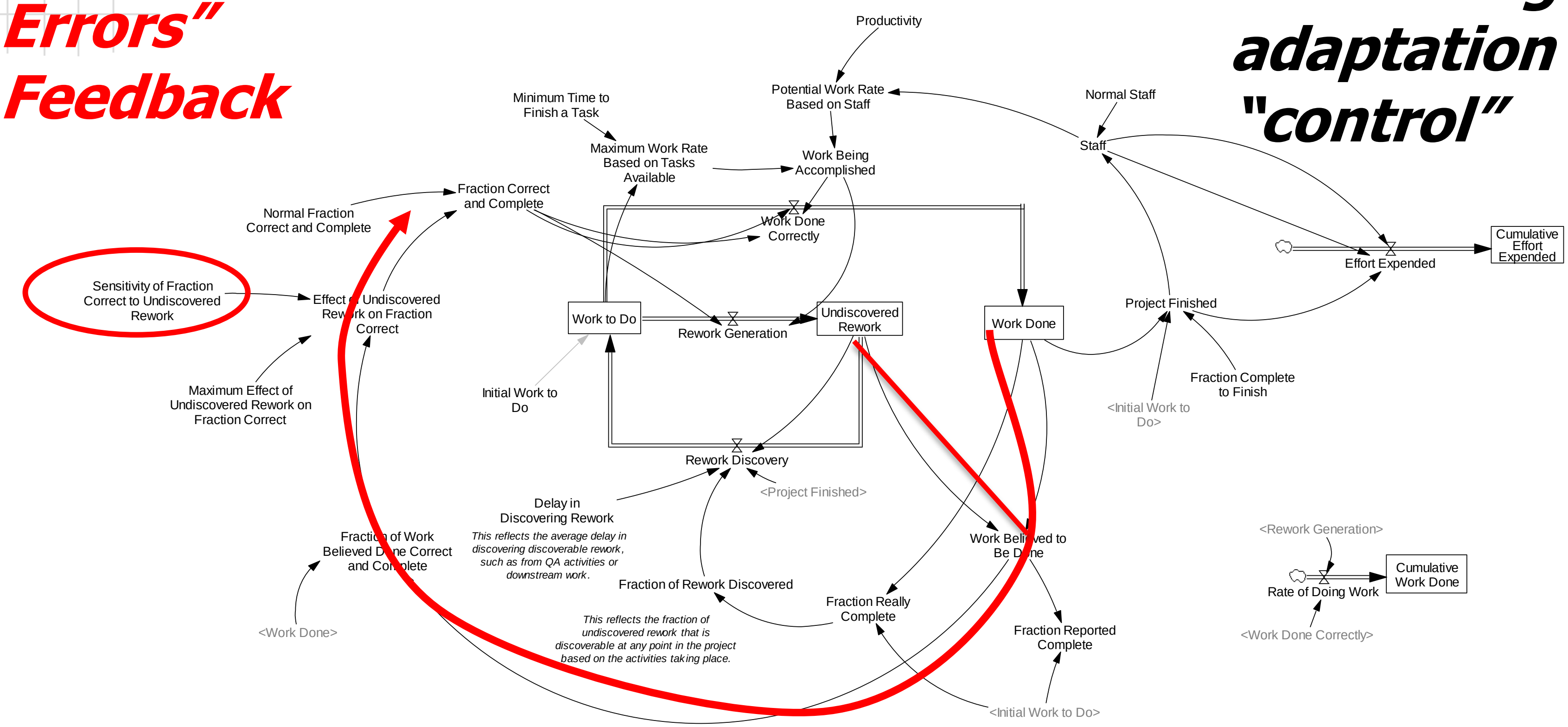




Complete Simple Model 1

"Errors on Errors" Feedback

No project monitoring or adaptation "control"



Representing Effects on Productivity and Fraction Correct

Productivity --

- PRODUCTIVITY =

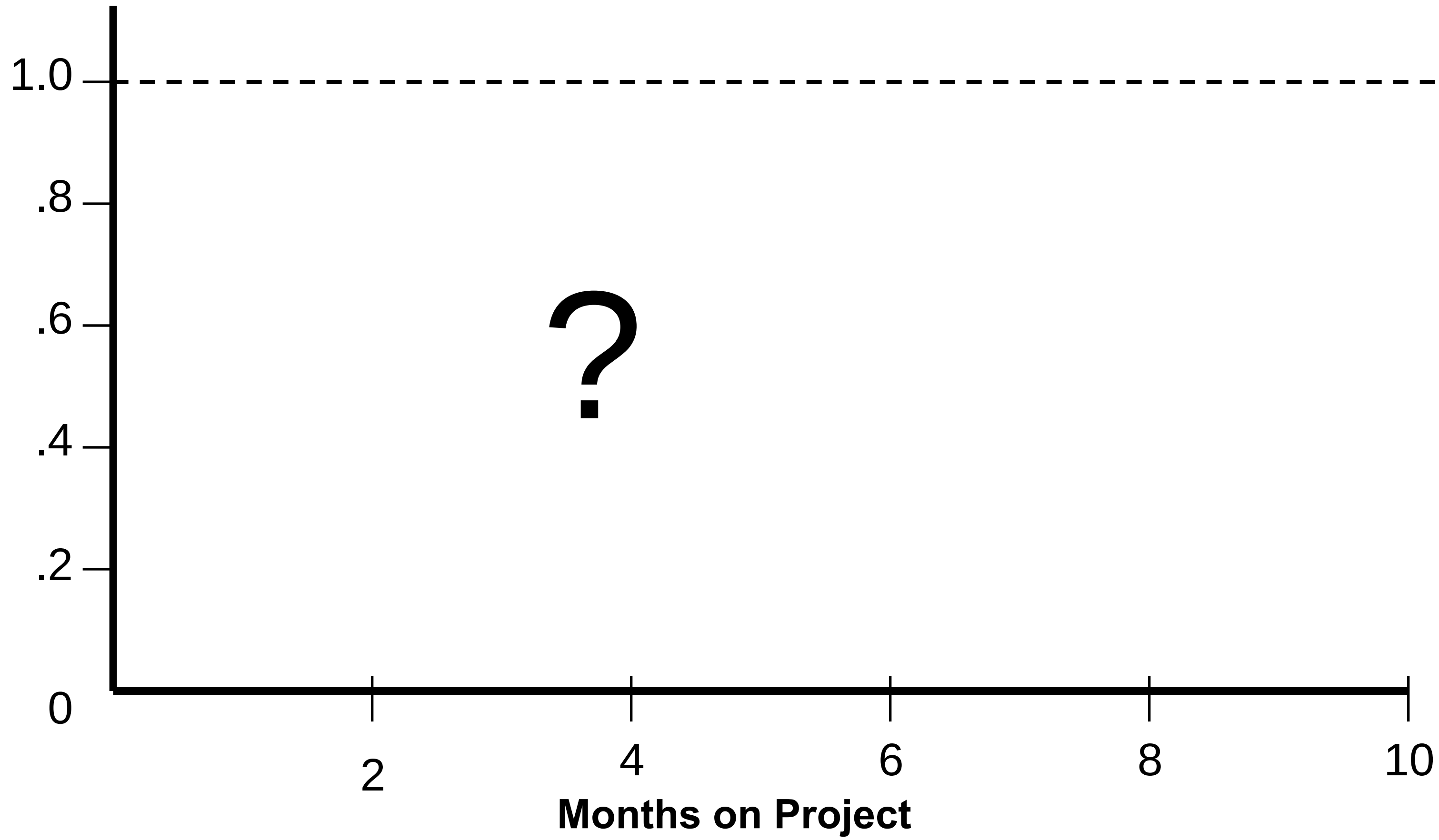
- Dimensions:
  - Productivity --
  - Normal Productivity --
  - Effects --

Fraction Correct --

- Fraction Correct = Normal Fraction Correct * Effect of Staff Experience * Effect of Undiscovered Rework* ...
- Dimensions:
  - Fraction Correct -- Fraction
  - Normal Fraction Correct -- Fraction
  - Effects -- “Dimensionless”

Effect of Staff Experience

Effect of Experience = Function (Months on Project)

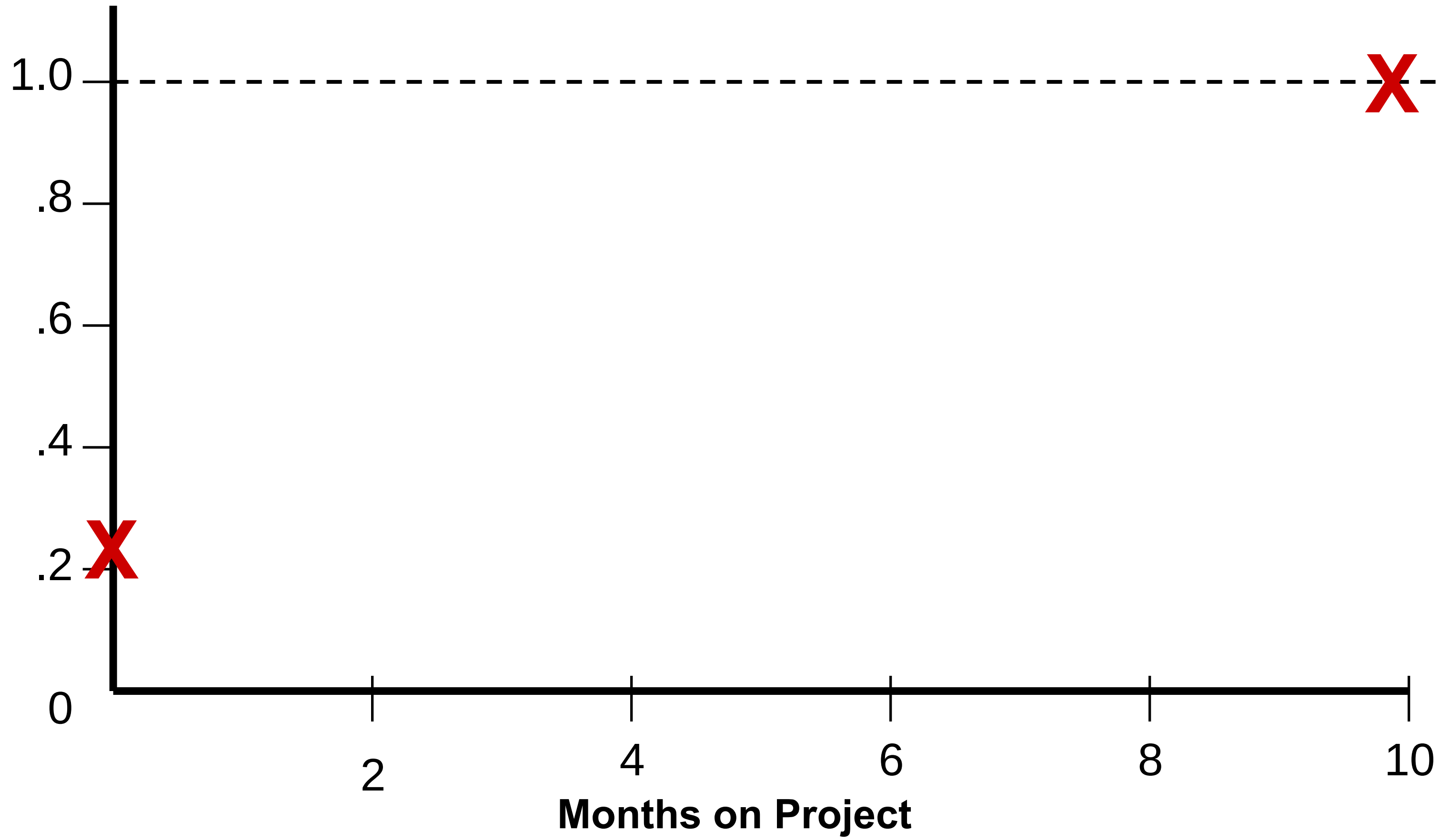


How do we determine these effects?

- The effects are first estimated based on "common sense."
- Specify likely values at extreme points, and draw a smooth curve in between
- Later, effects verified via model calibration and/or sensitivity testing.

Effect of Staff Experience

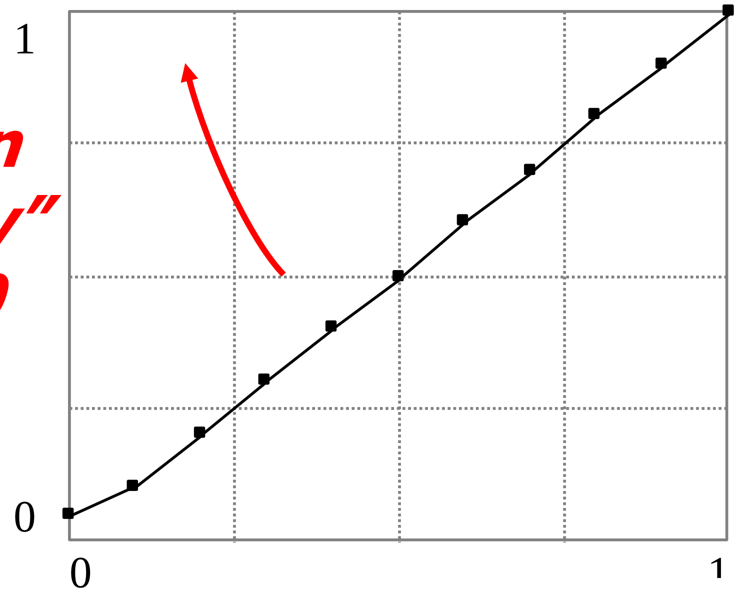
Effect of Experience = Function (Months on Project)



Effect of Undiscovered Rework on Fraction Correct:

Effect of Undiscovered Rework on Fraction Correct

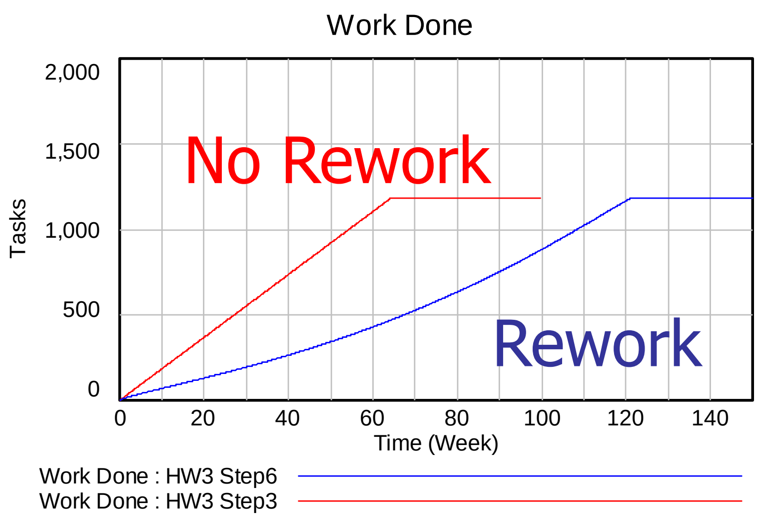
Decrease in "Sensitivity" from 1 to 0

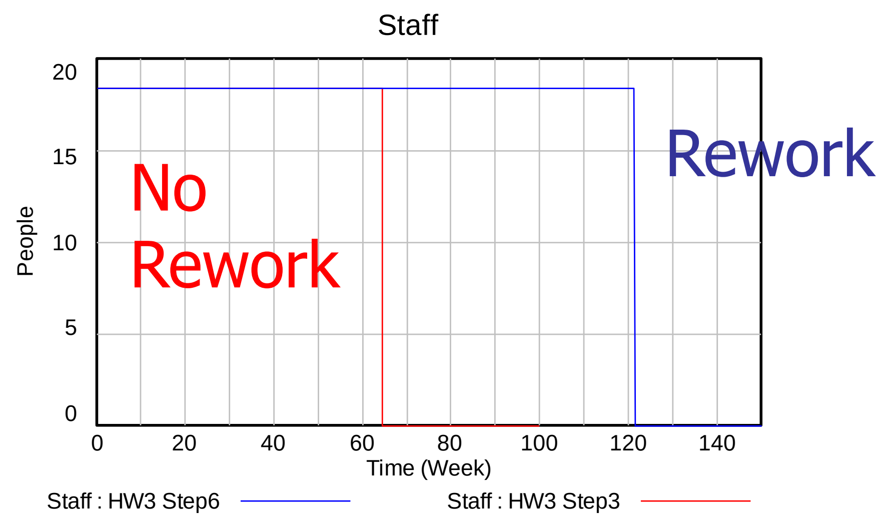


Fraction of Work Believed Done Correct and Complete

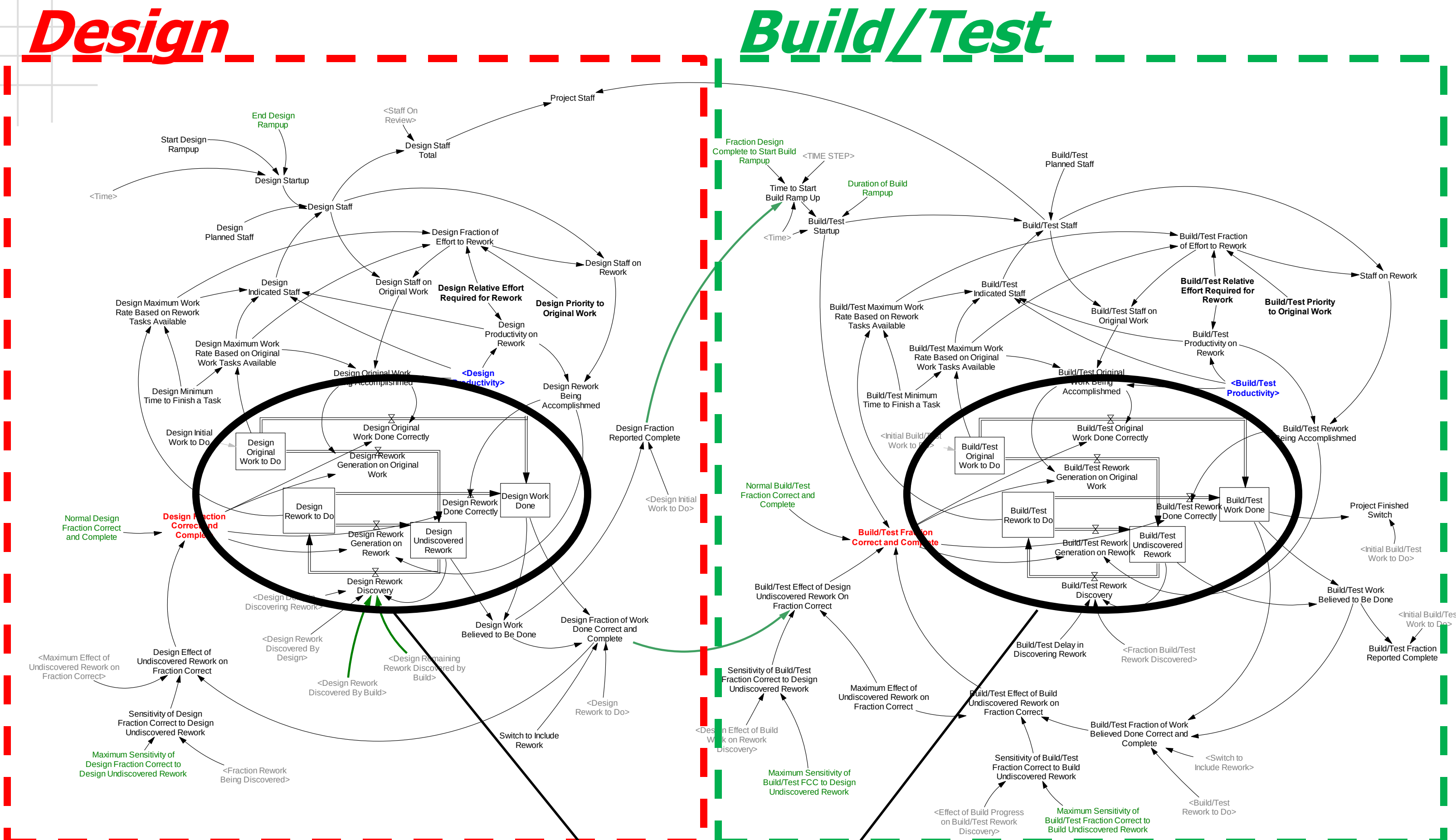
Note: The effect of undiscovered rework on fraction correct is assumed to be proportional -- an error in past work creates an error in current work. Given that in this simple model fraction correct represents several effects of work errors, this strong relationship may be reasonable.

HW#3 Part 1 *CityCar* Simulation





Project Model 2: Work Flows & Staffing in "Simple" Two Phase Model



Model you are given in Part 2 of HW#3

Four Stock Rework Cycle

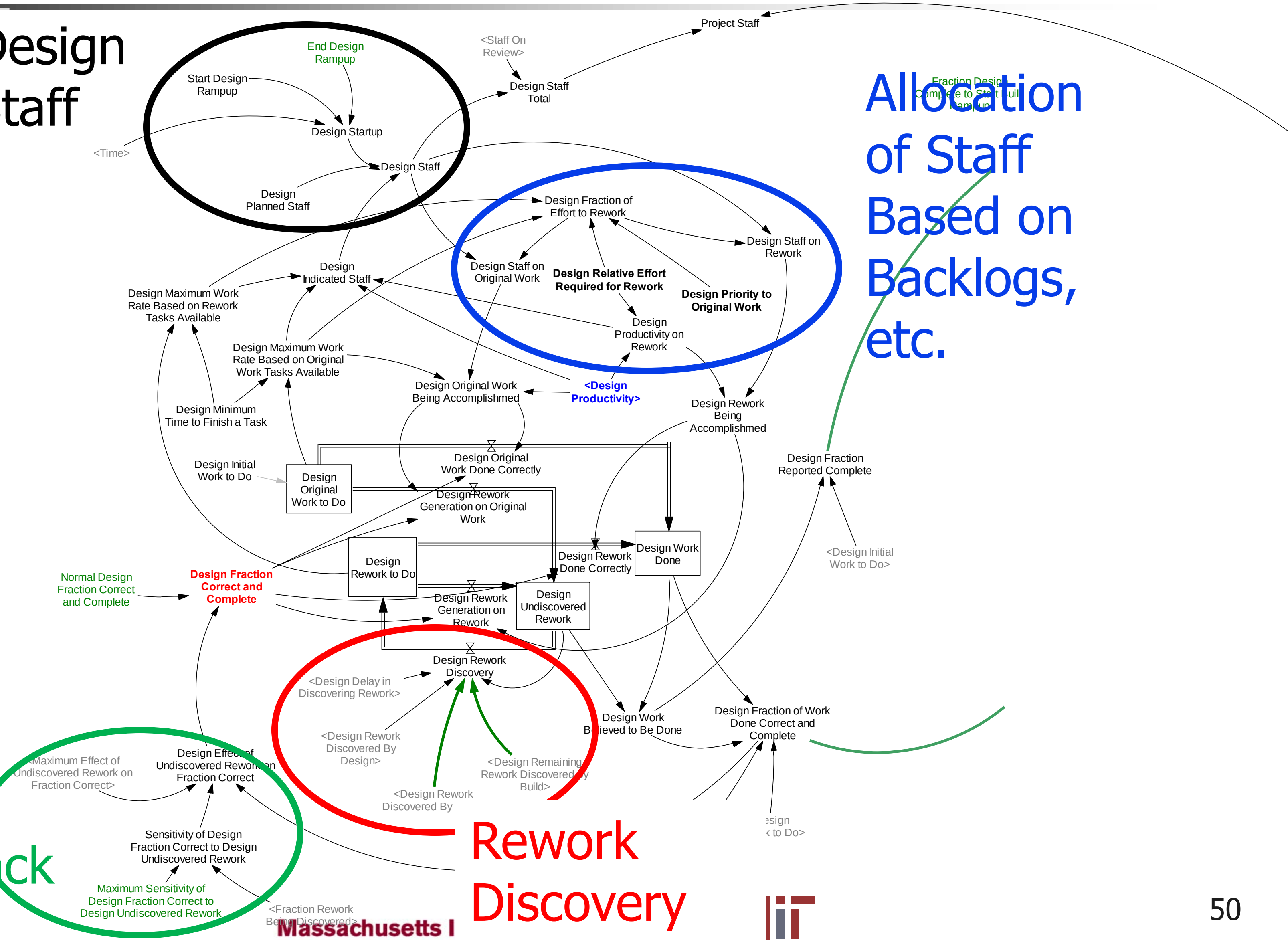
Design Phase of Work (Build/Test Similar)

Design Staff

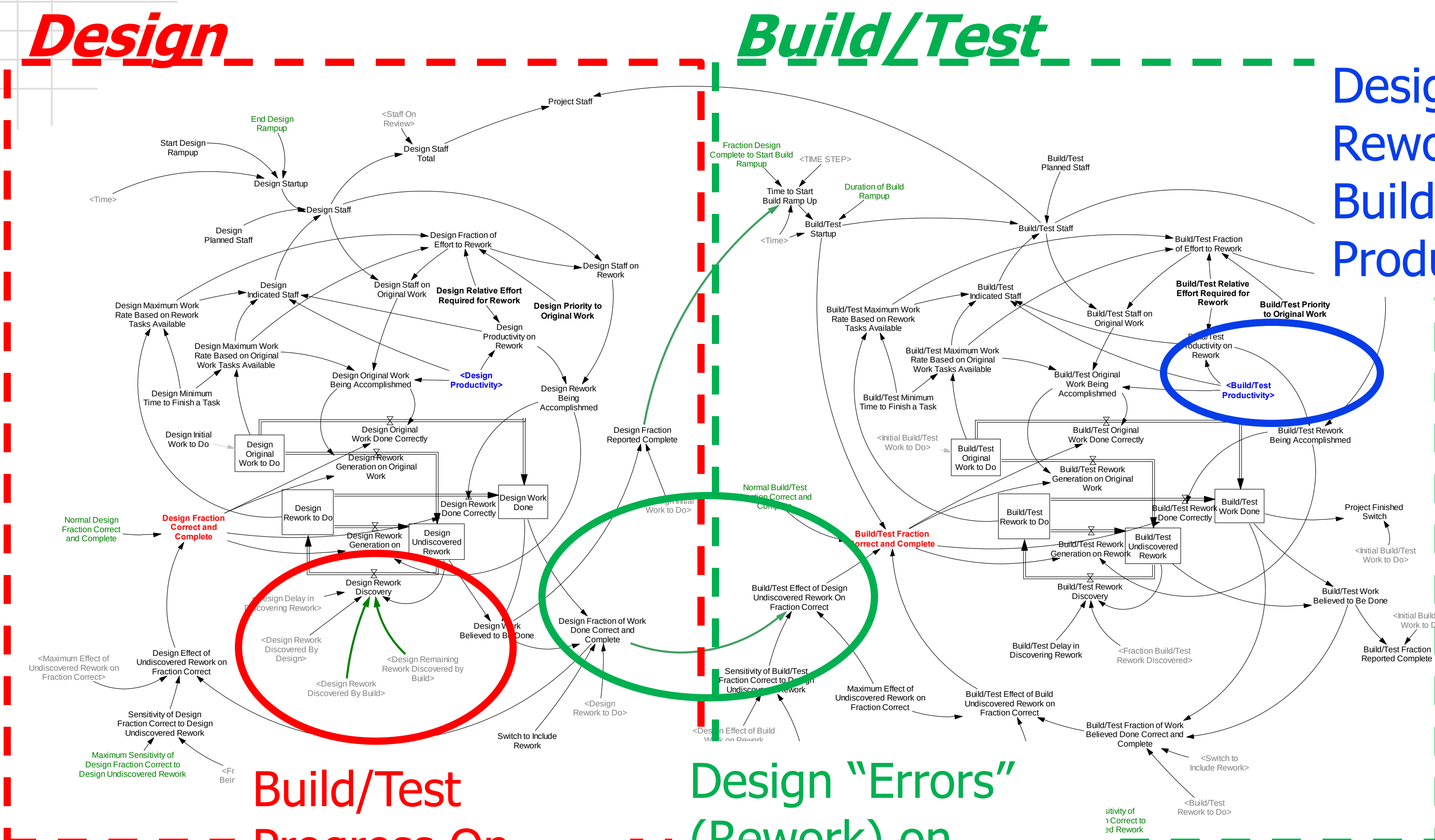
Allocation of Staff Based on Backlogs, etc.

"Errors Create Errors" Feedback

Rework Discovery



Phase Interconnections



Design Rework on Build/Test Productivity

Build/Test Progress On Design Rework Discovery

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

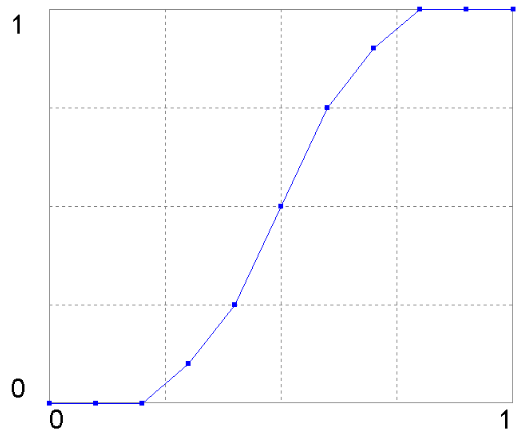
Key Parameters

	Design	Build/Test
Normal Productivity (Tasks/Week/Person)	1	1
Normal Fraction Correct and Complete (Fraction)	0.6	0.95
Relative Effort Required for Rework	0.5	0.5
Priority to Original Work	1	1
Max Sensitivity of Fraction Correct to In-Phase Undiscovered Rework	0.5	0.75
Max Sensitivity of Fraction Correct to Inter-Phase Undiscoverd Rework	NA	1
Max Sensitivity of Productivity to Undiscovered Rework	1	1
Max Fraction of Design Rework Discoverable by Design Work	0.6	NA
Fraction of Design Rework Disovered Based on Planned Iterations and Reviews	0.3	NA
Fraction Design Complete to Start Build Ramp-up	NA	0.99
Duration of Ramp-Up	1	1

Rework Discovery

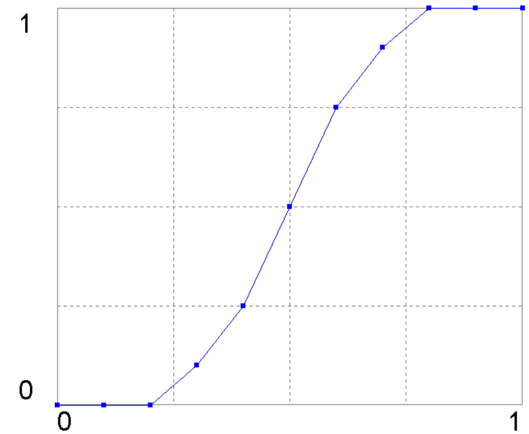
Max Fraction Discoverable in Design = 0.6

Graph Lookup - Effect of Design Progress on Design Rework Discovery



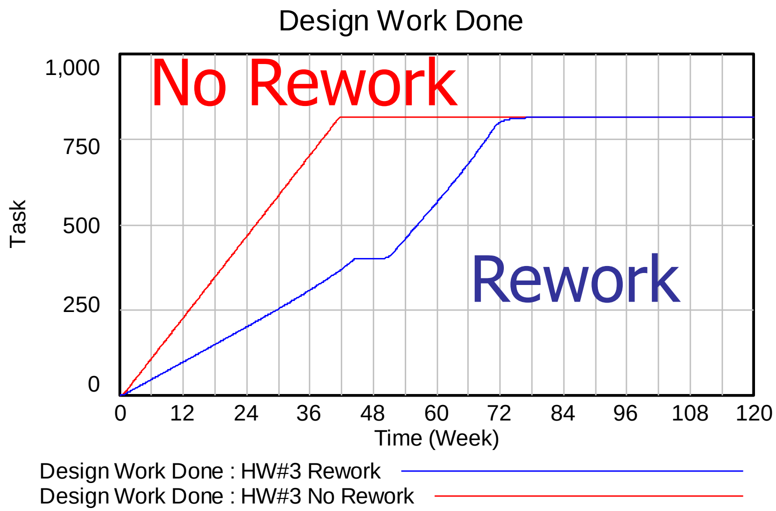
Design Fraction Original Work Complete

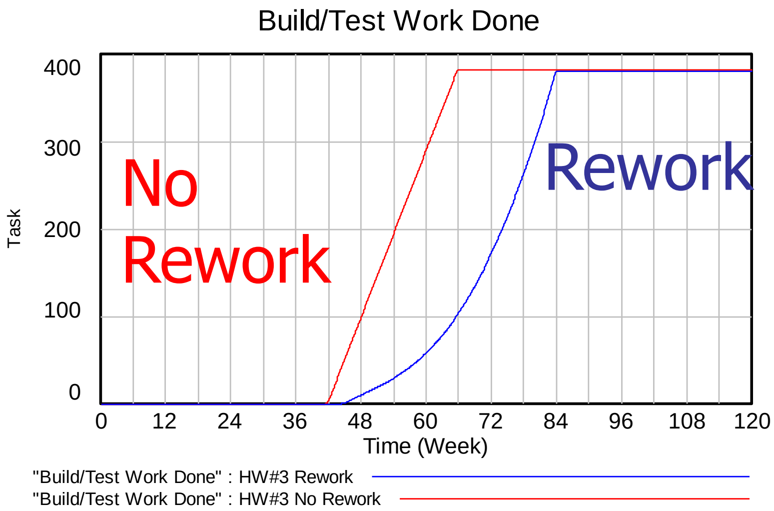
Graph Lookup - Design Effect of Build Work on Rework Discovery



Build/Test Fraction Original Work Complete

HW#3 Part 2 *CityCar* Simulation

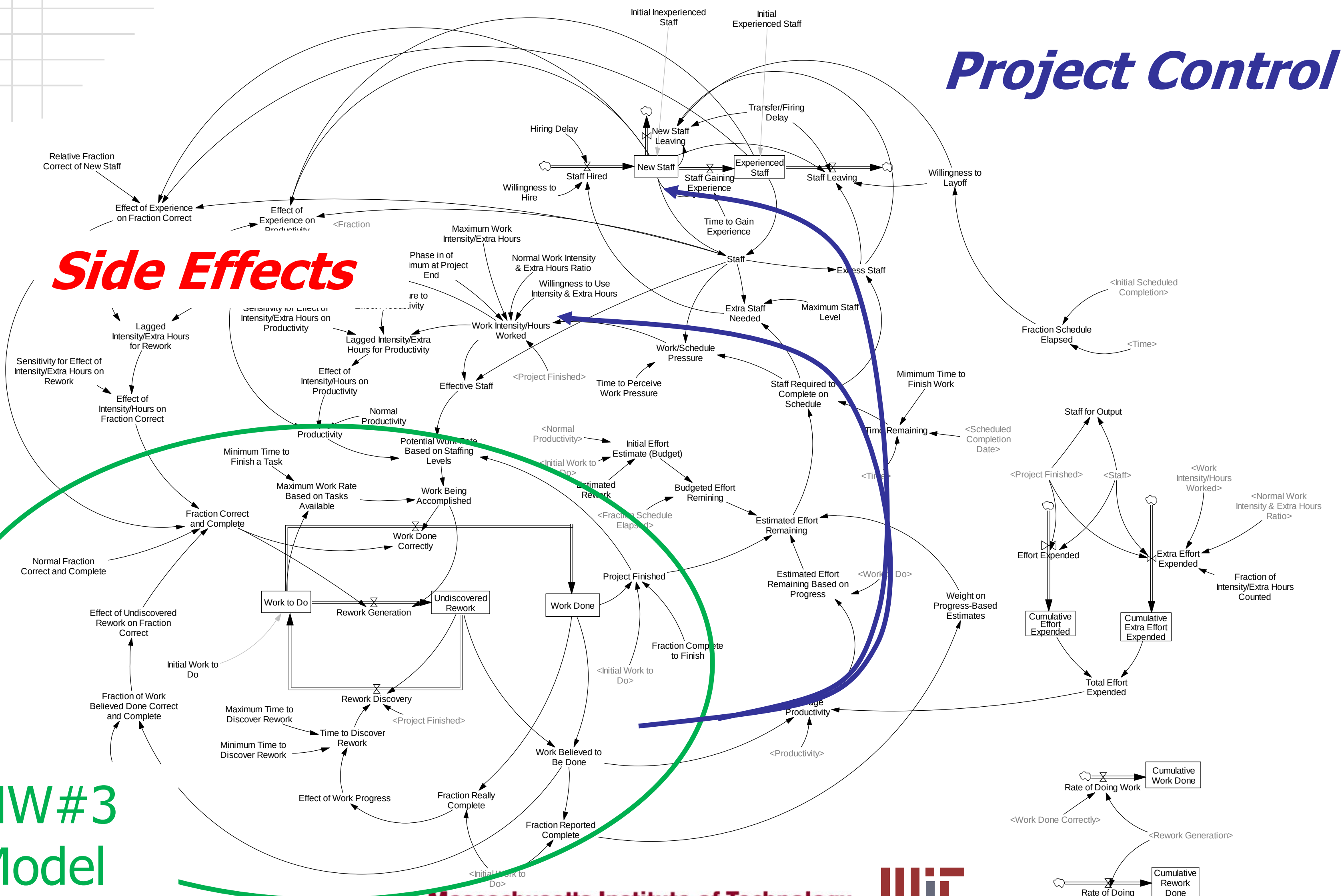




Project Model 1: Elaboration

- 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

Where we are headed – Full One-Phase SD Model !!



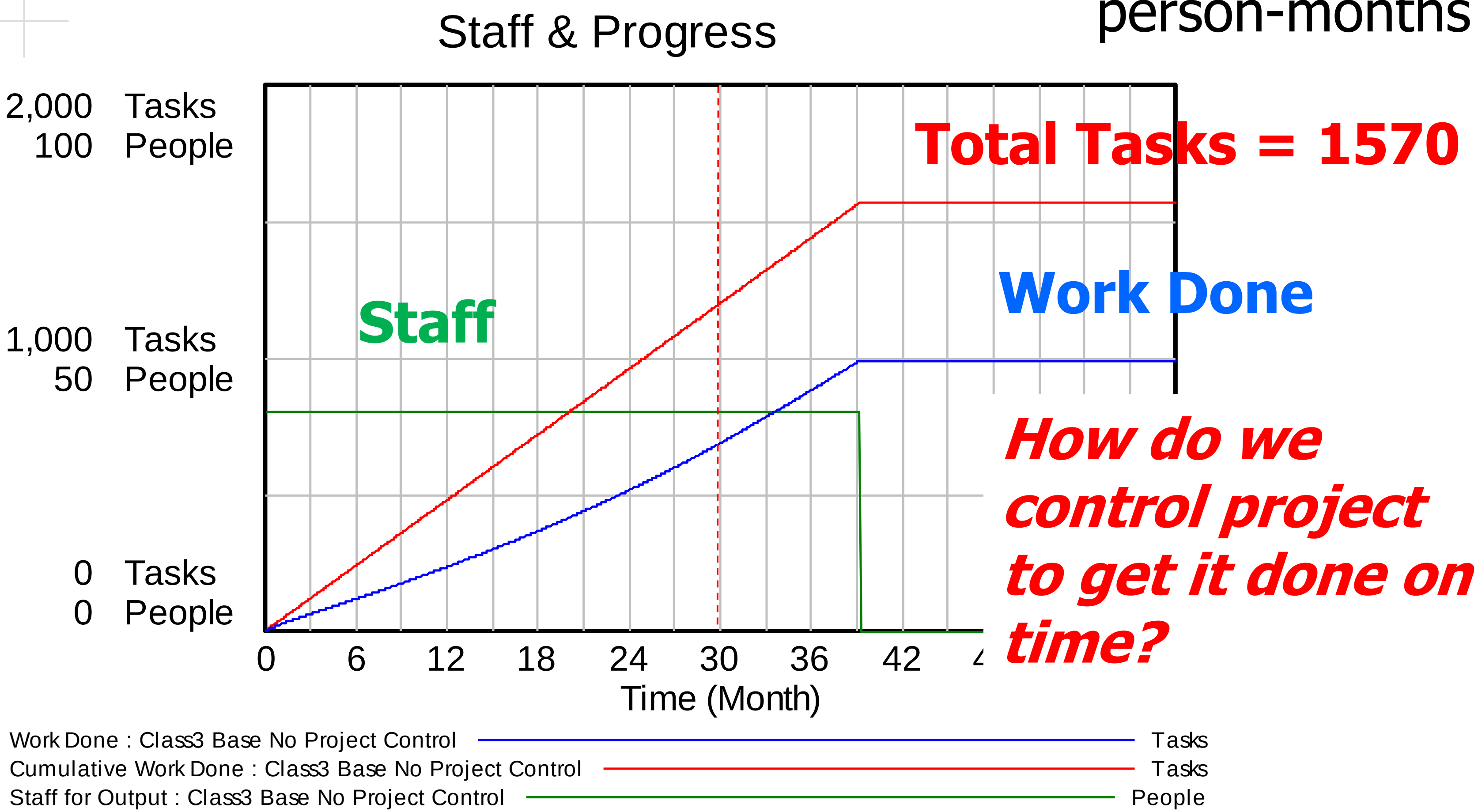
Example Project

- Scope = 1000 Tasks
- Scheduled Completion Date = 30 (Month)
- Staff = 40 (Implied budget of 1200 person- months, including 200 tasks estimated rework)
- Normal Quality = 0.85
- Productivity = 1 task/month/person

Note: Infeasible Plan

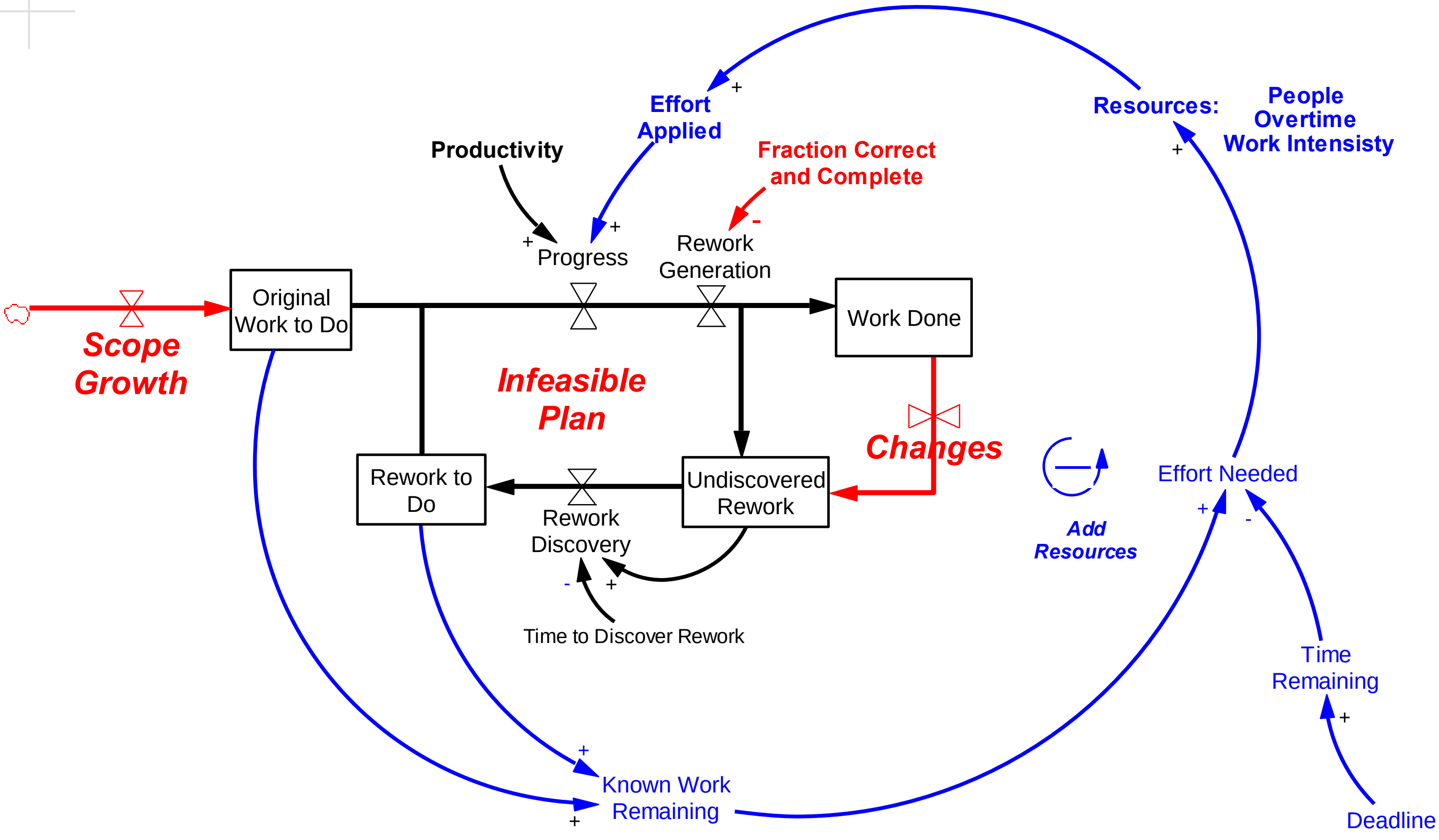
Project Behavior

Cost = 1570 person-months



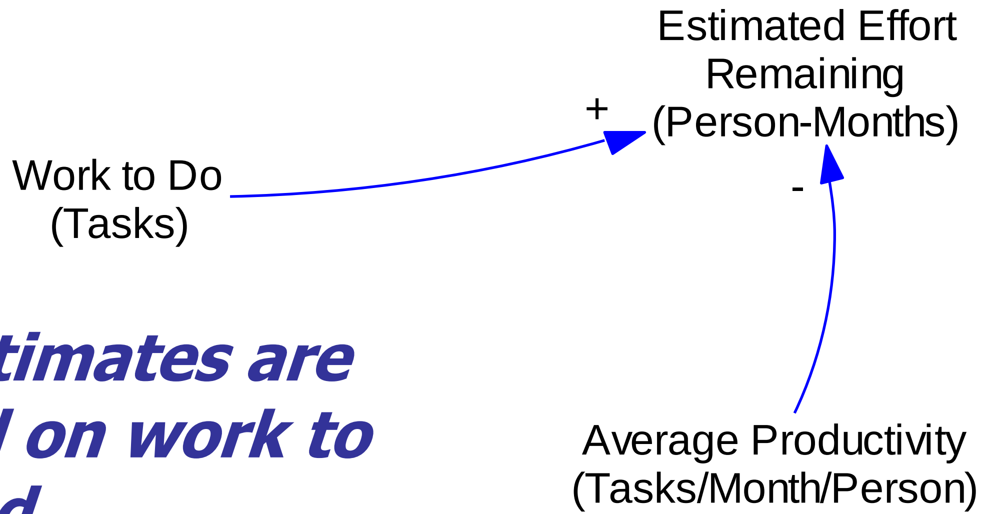
How do we control project to get it done on time?

Qualitative model representation



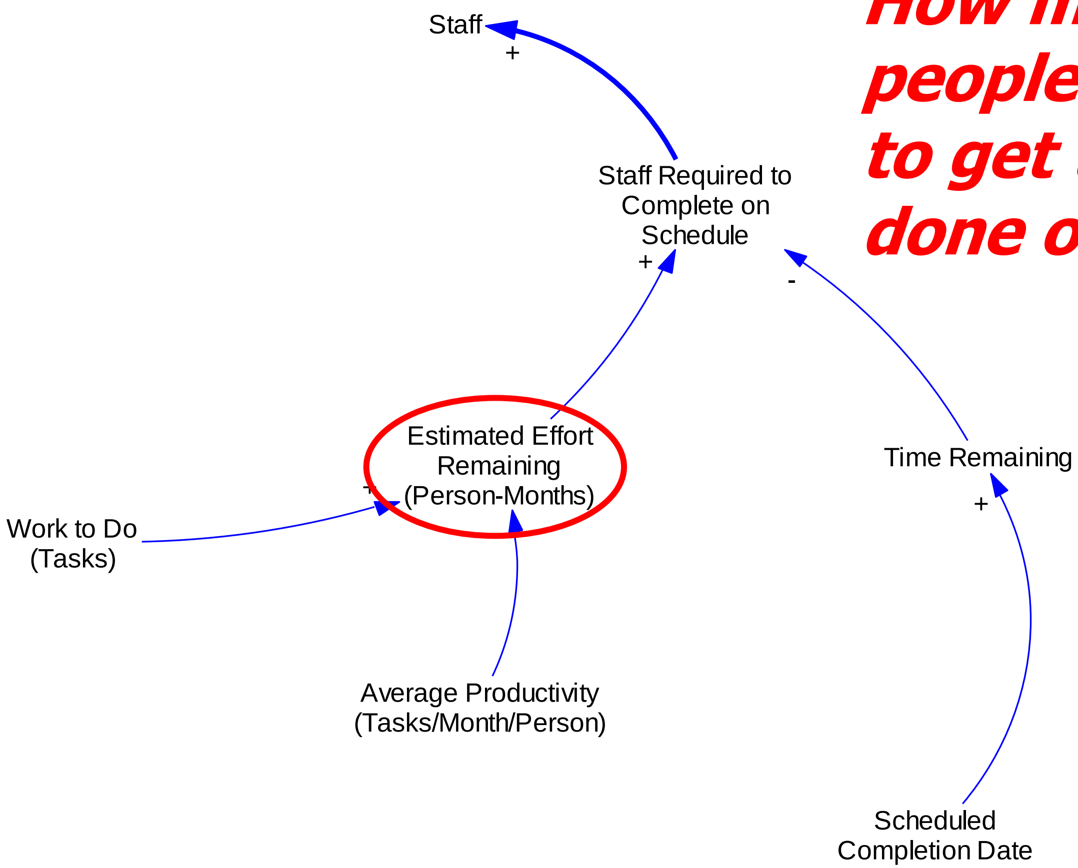
Project Control

1. Project control is driven by estimates of how much effort is left ...



2. Estimates are based on work to do and productivity (undiscovered rework?)

Project Control -- Staffing



How many people do I need to get the job done on time?

Staff Required = ***Estimated Effort Remaining*** / **Time Remaining** *[People]*

Project Control – Schedule

How many people do I need to get the job done on time?

When Can I finish with the current staff?

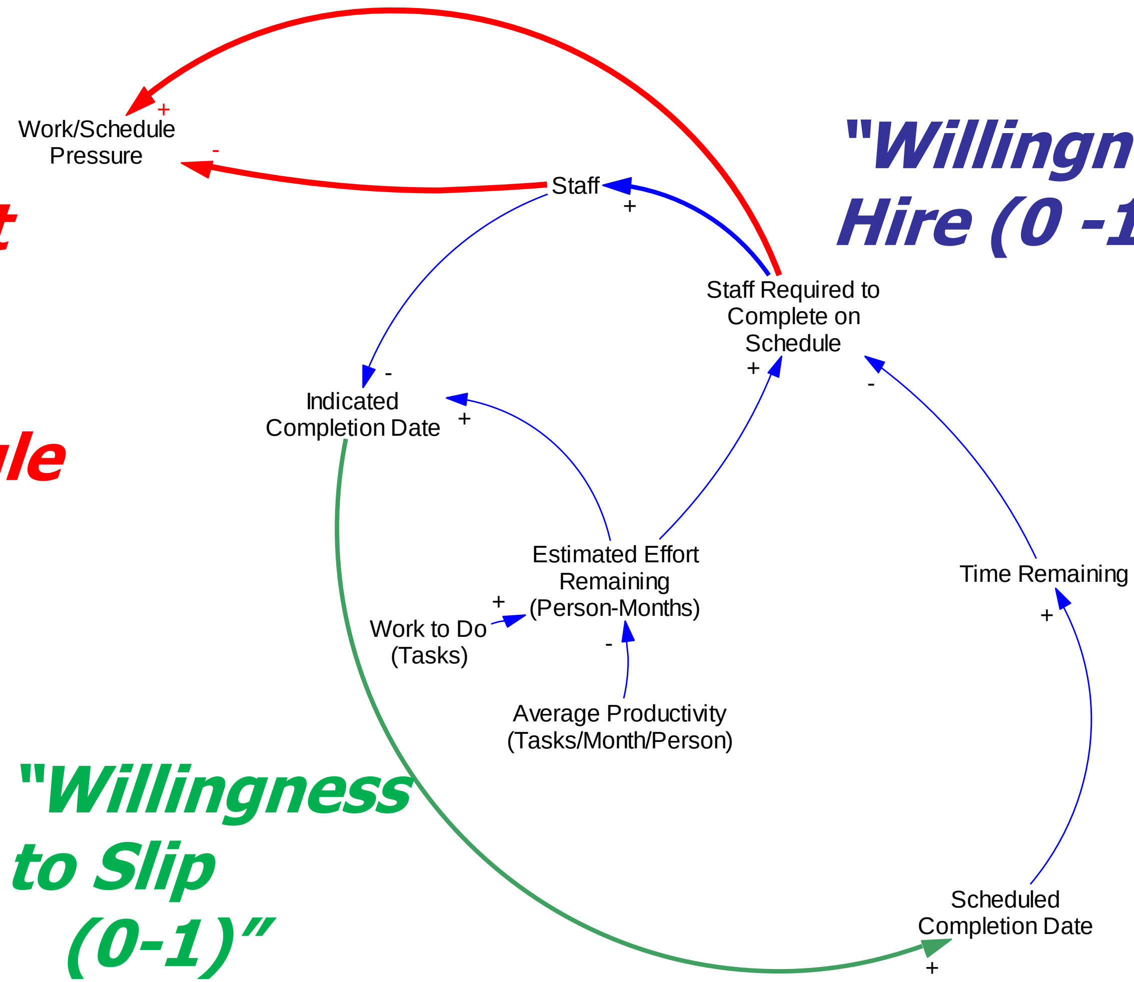
Indicated Completion Date = Time + (***Estimated Effort Remaining***/**Staff**) *[Month]*

Project Control

Based on Staff Required and Indicated Completion Date, three options:

1. Add Staff
2. Explicitly Slip Schedule
3. Exert "Schedule Pressure" (Work Intensity and Extra Hours)

Actions Determined By …



"Willingness to Hire (0 -1)"

To the extent these do not sum to 1, work/schedule pressure is residual.

"Willingness to Slip (0-1)"

Work / Schedule Pressure

Work/Schedule Pressure

- If a project falls behind schedule and staff are not added or schedule slipped, management …
  - Pressures team to work faster
  - Team works longer hours/overtime

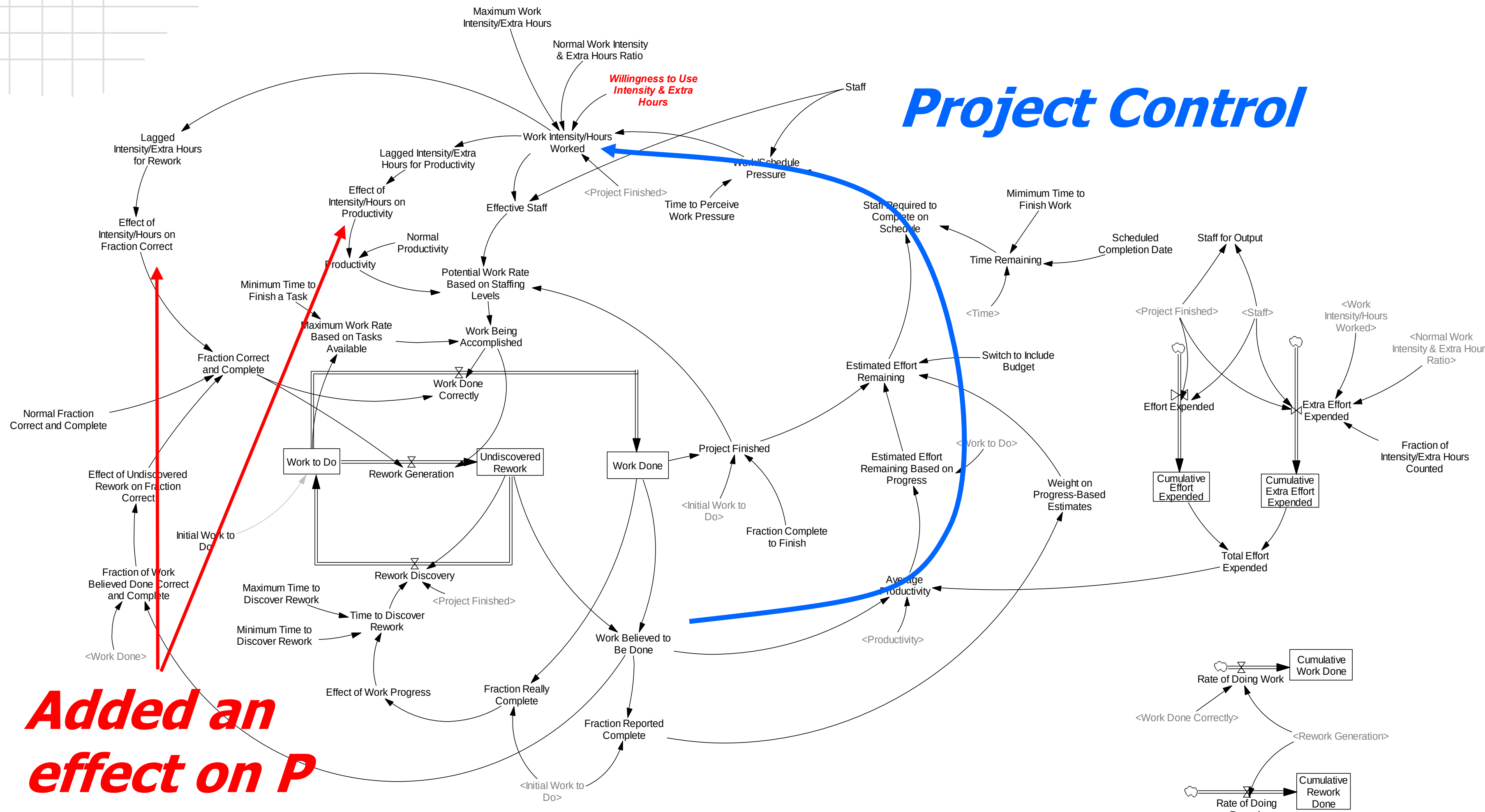
➔ Represented as impact on "effective staff" (= staff * intensity-overtime ratio)

Schedule Pressure

- Downsides (Side Effects) --
  - "Haste makes waste"
  - Fatigue adds to mistakes (and may reduce productivity)

➔ Represented as impact on fraction correct and complete, and on productivity

Intensity/Extra Hours Loops Added to Model

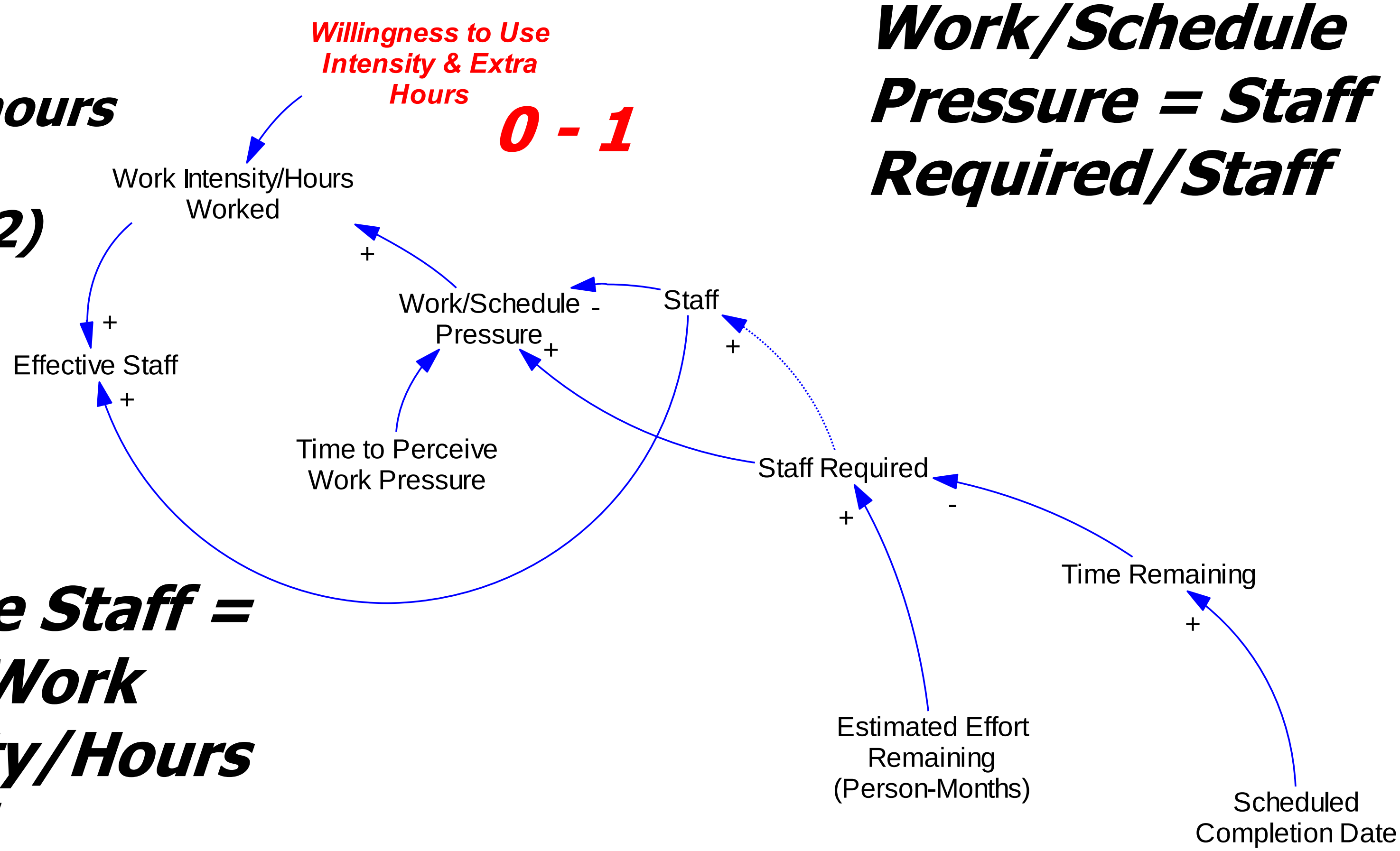


Project Control

Added an effect on P and FCC

Work/Schedule Pressure

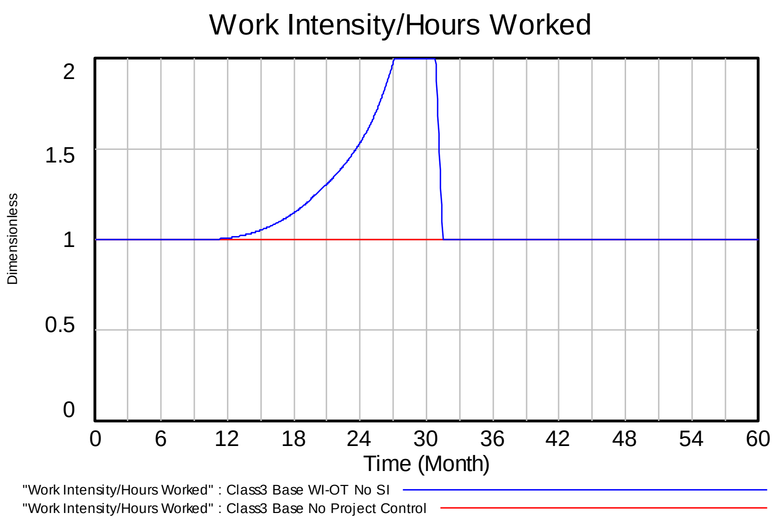
Subject to maximum intensity/hours worked (currently 2)

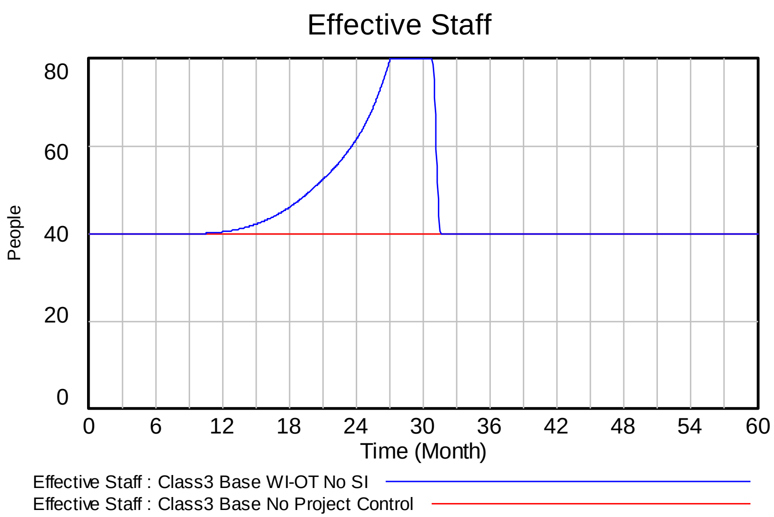


Work/Schedule Pressure = Staff Required/Staff

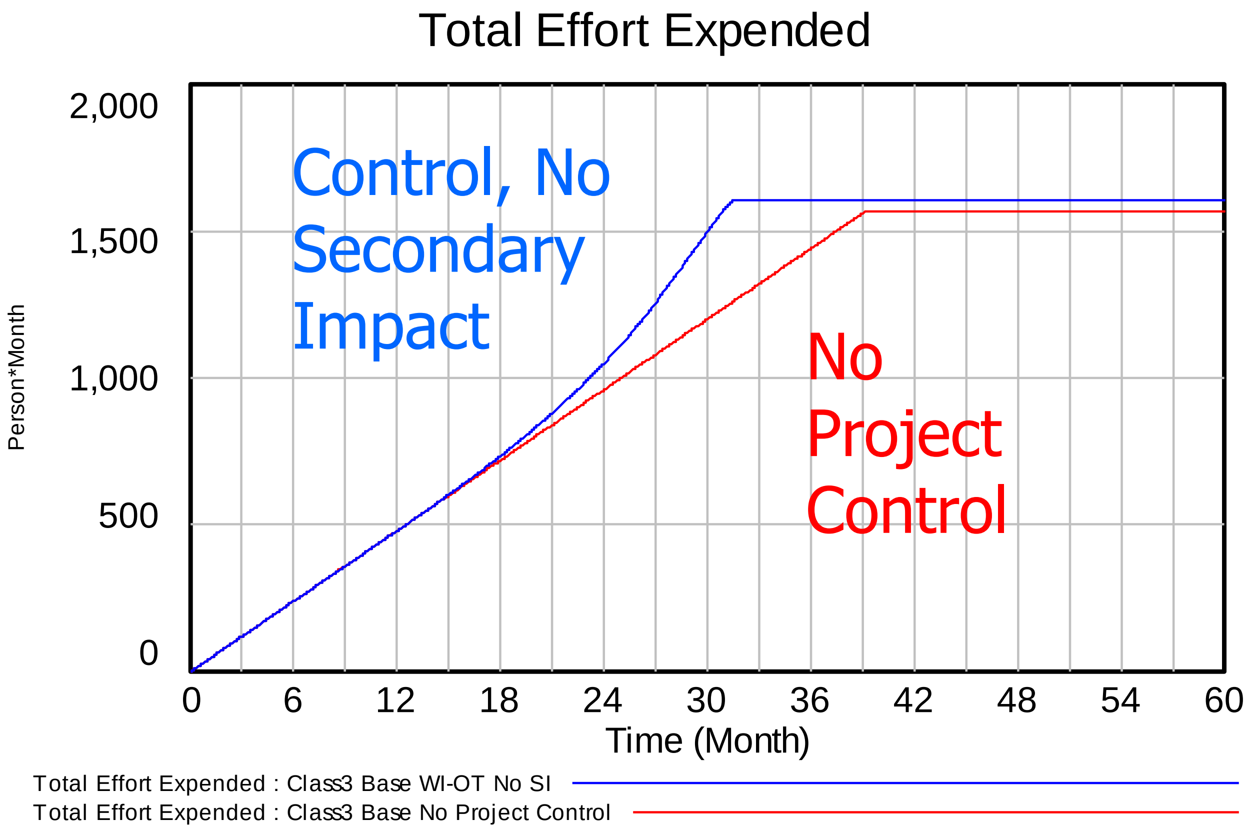
Effective Staff = Staff * Work Intensity/Hours Worked

Without Secondary Impact …

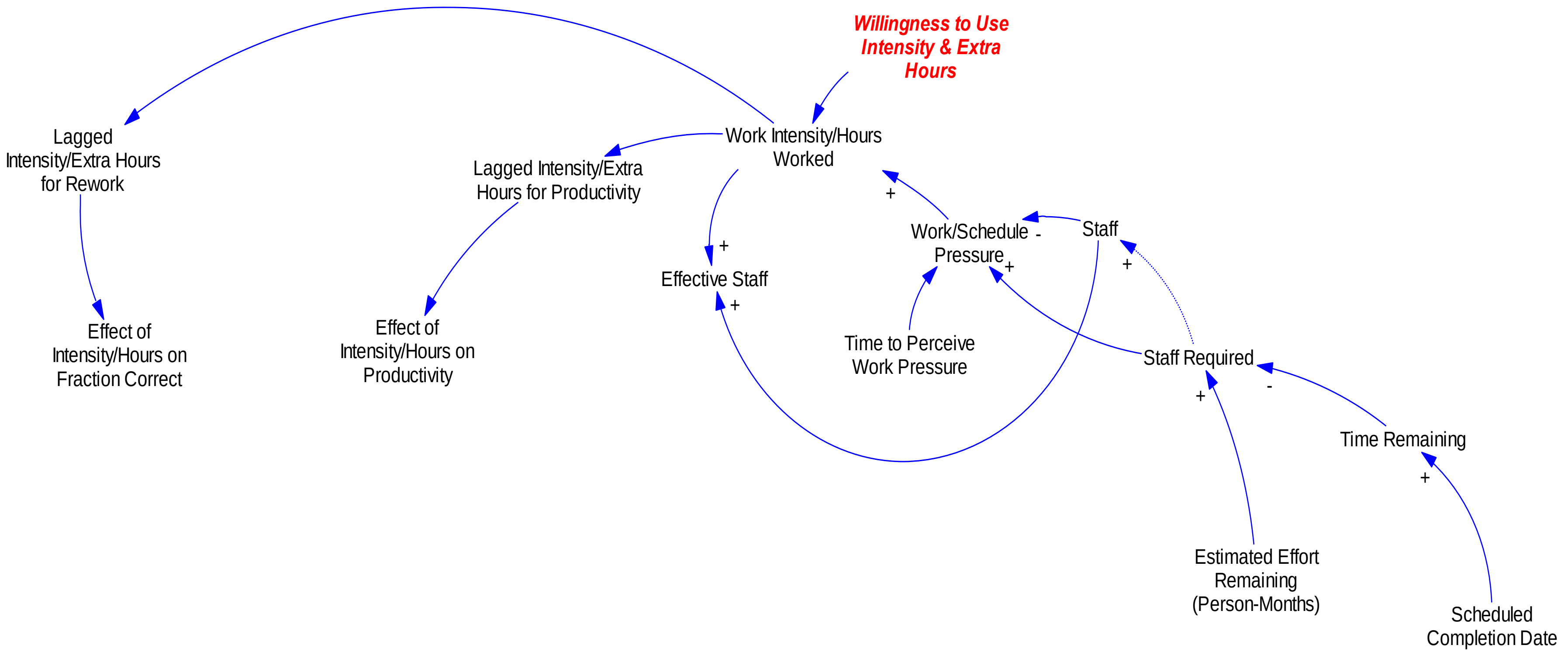




The project finishes sooner

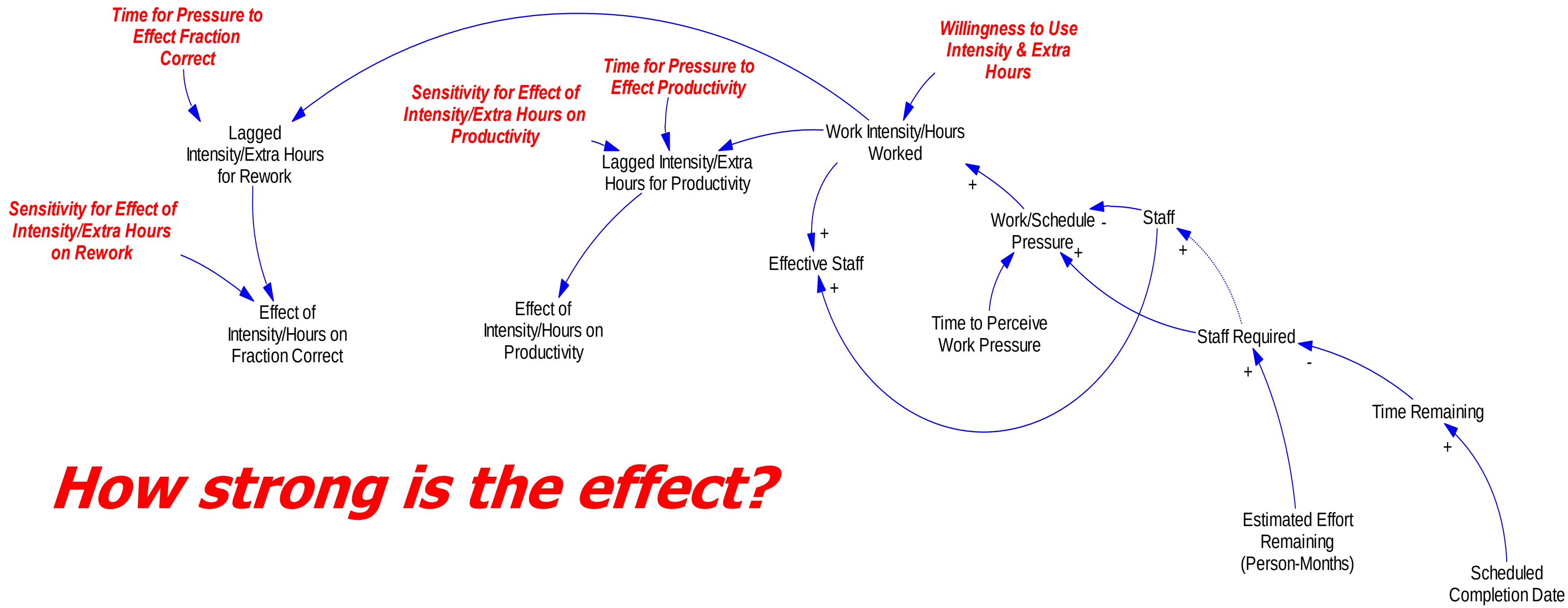


Impact on Productivity & Fraction Correct



Key Parameters: Impact Sensitivity and Delay

How long does it take before Intensity/Longer Hours affects Productivity and Fraction Correct?



How strong is the effect?

Impact determined by "sensitivity"

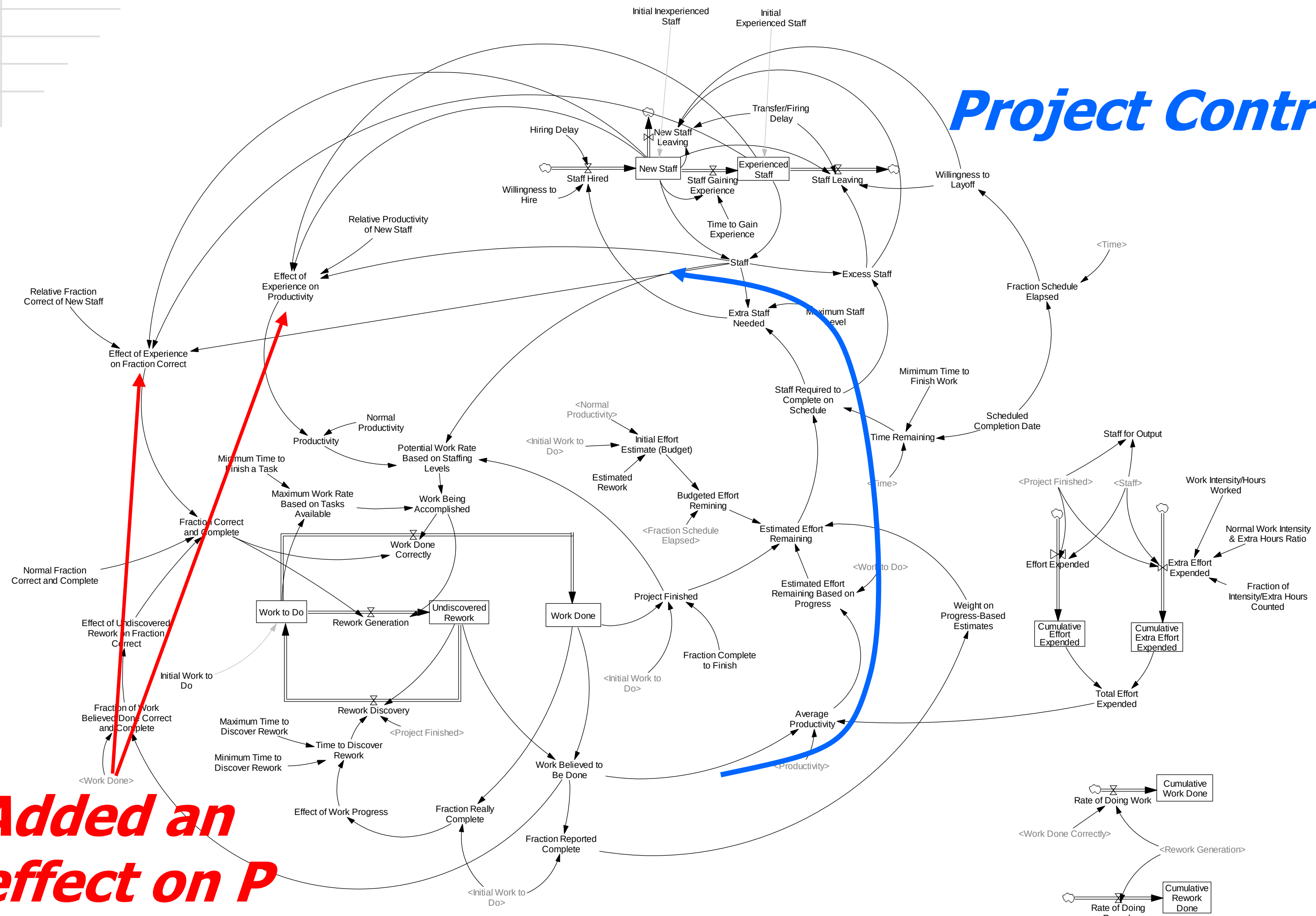
- Productivity
  - Sensitivity =1 ➔ productivity falls such that *additional* output is zero
  - Sensitivity = 0 ➔ no reduction in pdy
  - Sensitivity = 0.5 ➔ *additional* output 50%
- Fraction Correct
  - Sensitivity =1 ➔ all *additional* output contains errors
  - Sensitivity = 0 ➔ no reduction in fraction correct
  - Sensitivity = 0.5 ➔ *additional* output 50% errors

See text for implementation details (SD3.4.2, pp. 27-34).

Staffing

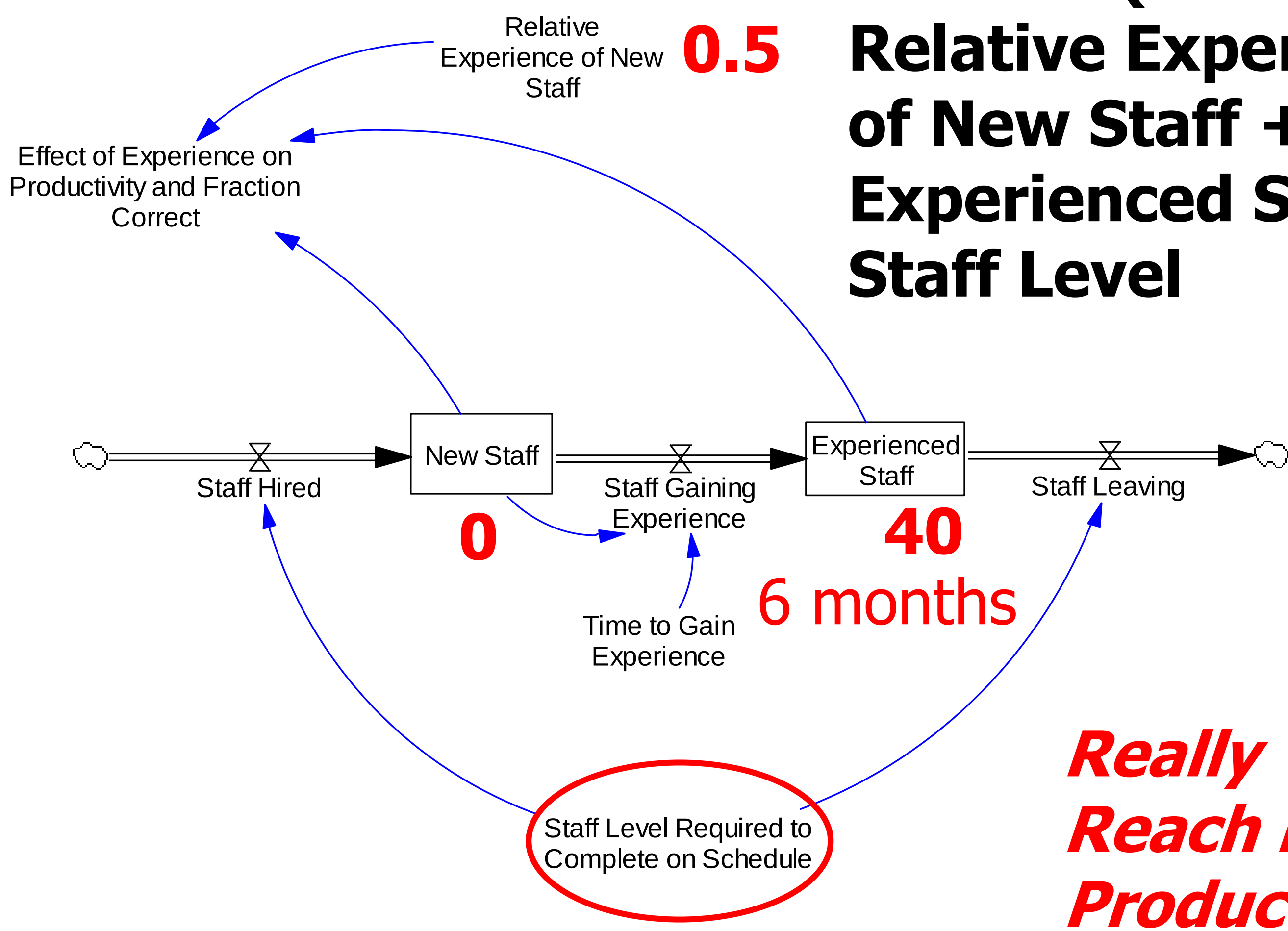
Staffing Added to Model

Project Control



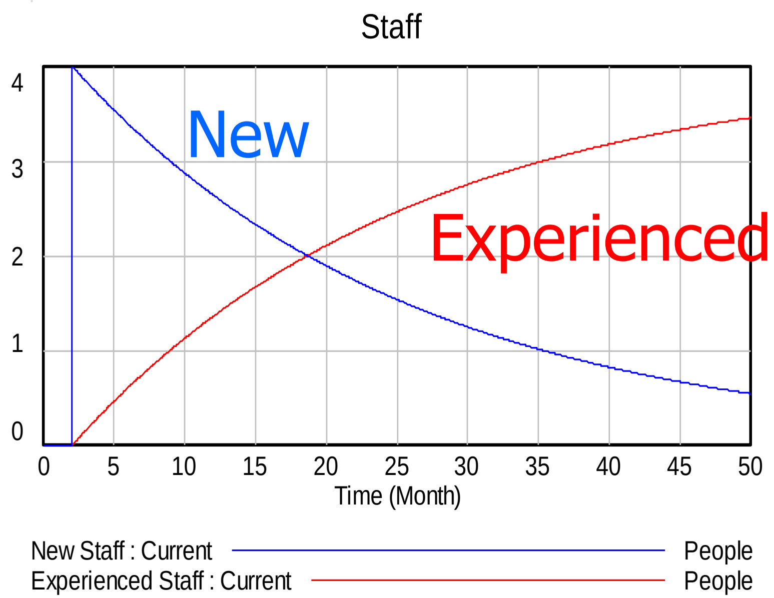
Added an effect on P and FCC

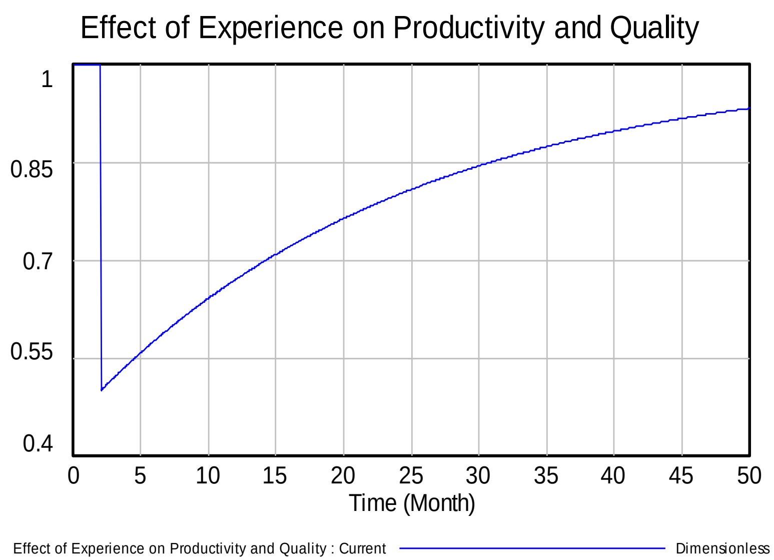
Effect = (New Staff * Relative Experience of New Staff + Experienced Staff) / Staff Level



Really "Time to Reach Full Productivity"

Adding 4 New Staff at Time 2

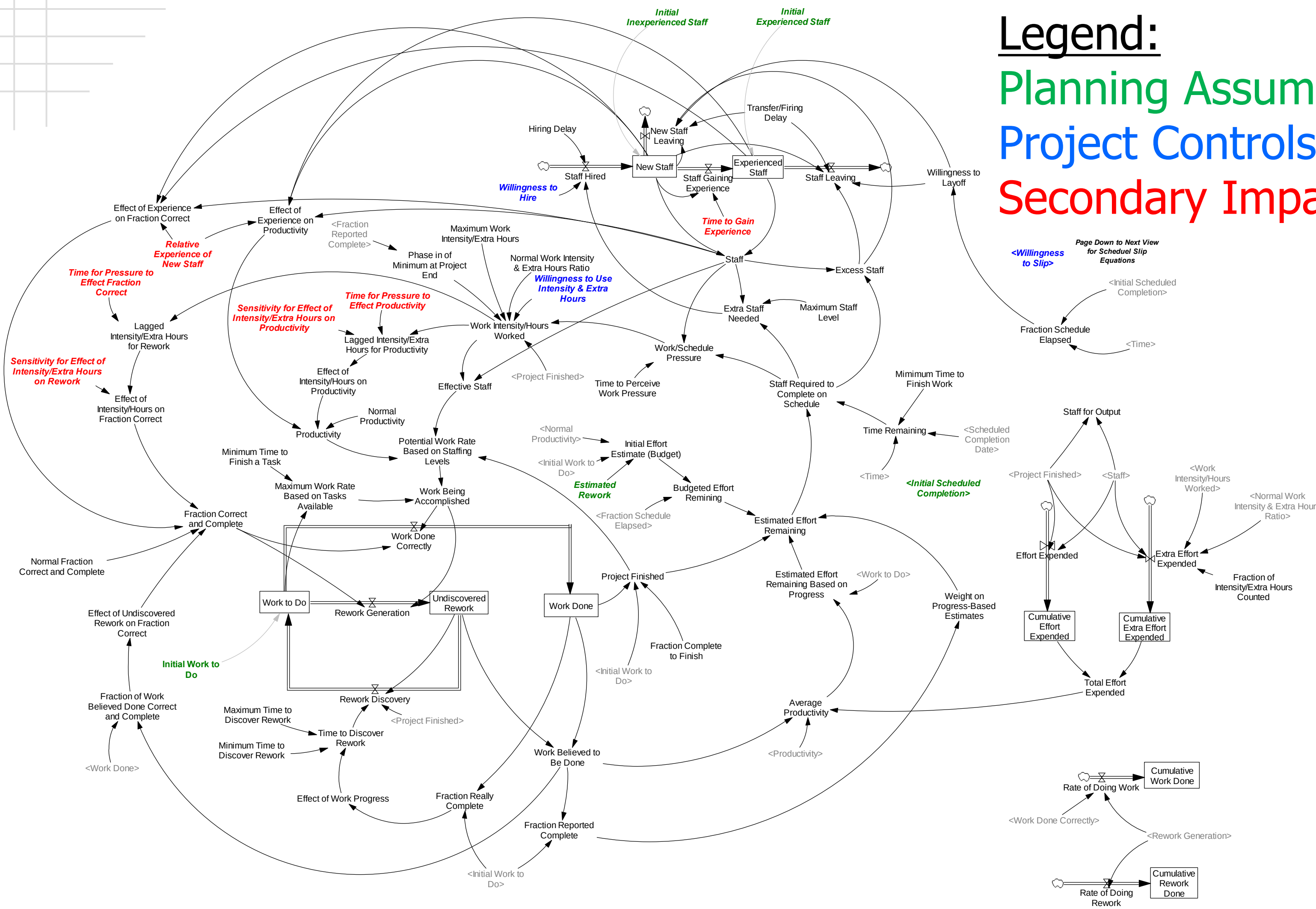




- We'll discuss simulations of secondary impact in two weeks and in HW#5.

Changing Schedule (see textbook)

Full SD Model (Chapter 3)



Legend:

Planning Assumptions

Project Controls

Secondary Impacts

Planning Assumptions

- Scope = 1000 tasks
- Estimated Rework = 200 tasks
- Scheduled Completion Date = 30 (Month)
- Staff = 40 (Implied budget of 1200 person- months, including 200 tasks estimated rework)
- Normal Quality = 0.85
- Productivity = 1 task/month/person

Project Controls

- Willingness to Hire
- Willingness to Slip
- Willingness to Use Intensity & Extra Hours

Note: Should add to 1.0?

Secondary Impacts

- Relative experience of new staff
- Time to gain experience
- Sensitivity for effect of intensity/extra hours on productivity
- Sensitivity for effect of intensity/extra hours on rework
- Time for pressure to effect productivity
- Time for pressure to effect rework

Today's Agenda

- Project Dynamics: Feedback Loops
- Qualitative Lessons
- Quantitative Models
- Validation and Model Extensions

Additional Issues

1. Model validation
2. Model extensions and elaborations

These are discussed in textbook Chapter SD3.

Model Validation

- Does the structure reflect what happens on projects?
  - Rework Cycle?
  - Staffing Dynamics?
  - Project Controls?
  - Effects on productivity and fraction correct?
- Are the parameters reasonable?

How do we know the "real" effects?

- Relative experience of new staff?
- Sensitivity of productivity and rework to overtime?
- Time delays for impact?

You're Uncomfortable With Quantifying All These Effects. What Are Your Options?

1. Ignore effects and estimate (simulate) impacts as if they did not exist
   - But that's the only value you know is wrong!
2. Use your experience/intuition/ "mental model" instead (no simulation)
   - I.e., try to account for effects simultaneously in your head that you can't do individually in a computer model
3. Use computer model with educated estimates …
   - Test sensitivity of results to exact values
   - Gather data and calibrate where warranted

Summary of ***Computer*** Model

- Three effects on Productivity & "FCC":
  - Errors on Errors
  - Work/Schedule pressure (represents overtime, fatigue, out-of-sequence work)
  - Experience (represents staff diversion & training, size of organization)
- Decisions to increase or reduce staffing
- Decisions to change scheduled completion date

Are There Alternative Models?

- More Productivity & “FCC” effects, etc.
- Variations on the basic rework cycle
- Multi-project and organizational models

More Productivity and Fraction Correct & Complete Effects:

- Model has 3 effects on P & Q :
  - Errors on Errors
  - Schedule pressure
  - Experience
- What additional affects could be included?
  - Morale
  - Overtime/fatigue
  - Sequence
  - Other types of experience
  - # projects/person
  - Organizational Size Changes
  - Availability of supplier information
  - and materials
  - Skills match to needs
  - ...

Other Types of Experience

A model might separately represent different drivers of experience, e.g.:

- Effect of Experience on Project = Function (Time on Project)
- Effect of Experience in Field = Function (Years Working)
- Effect of Skill = Function (Inherent Skill)
- Learning (Fraction Complete?)

Discussion – Representing Experience

- Experience on Project
  - Relative productivity = 0?
  - Time to gain experience fraction of project duration
- Experience as Engineer
  - Relative productivity = ?
  - Time to gain experience = duration of project or longer

Other Stocks

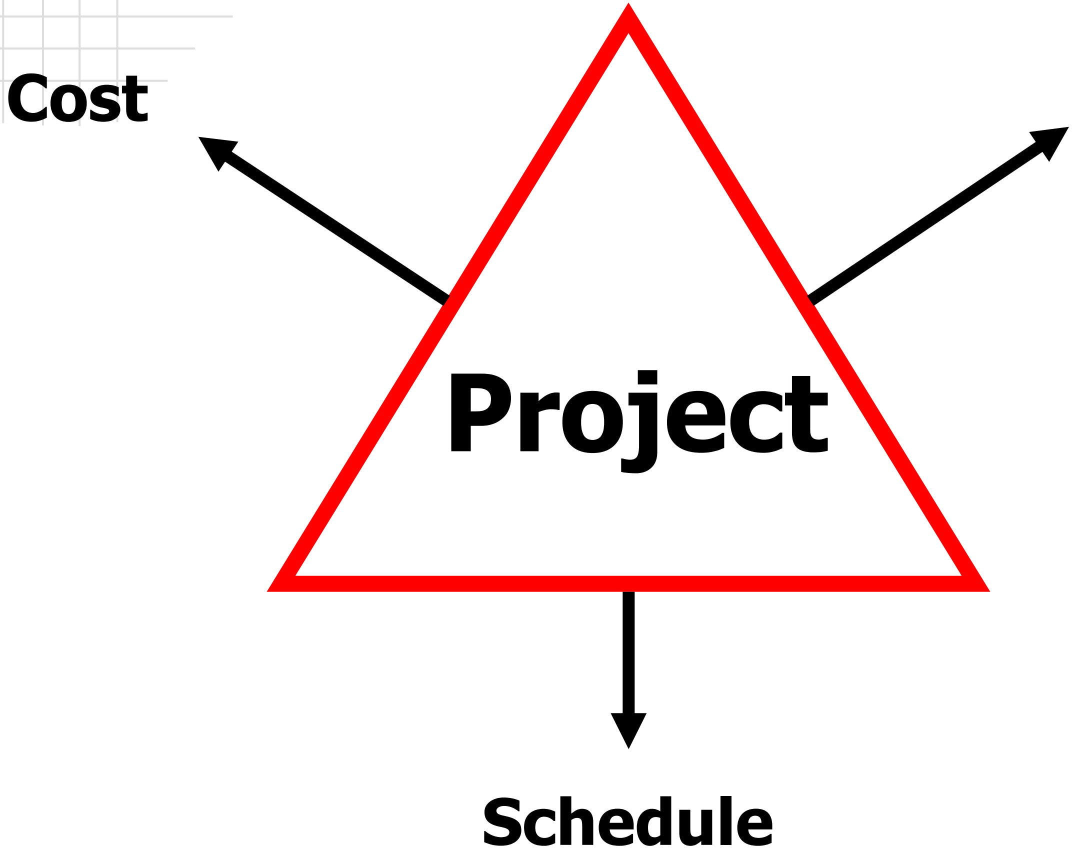
- Budget
- Knowledge
- Morale
- Technology
- Priority of Project
- Scope/features/customer needs
- Other resources

Other simplifications?

- Task dependence/sequence is not represented explicitly -- with enough staff, could finish the project in a week
- Only one phase of work explicitly represented
- Suppliers are not represented
- Interactions with other projects are not represented

These are treated endogenously or exogenously in more comprehensive models.

The "Iron Triangle"



What happens when something goes wrong?

Other Responses?

- To achieve target schedule …
  - ***Add resources***
  - Reduce scope
  - Ship with "bugs"
  - …
- To achieve target cost (total vs annual spend)
  - Reduce scope
  - Ship with "bugs"
  - Slip schedule to control annual spend
  - …
- To achieve target scope …
  - ***Add resources***
  - Slip schedule
  - …

Focus on achieving (1) scope & (2) schedule …

When is the project finished?

- In the current model, keep working until all work is completed correctly. In other situations, schedule may be more critical and therefore the project might:
  - reduce scope to meet schedule
  - ship with errors (undiscovered rework)

We will discuss other options later in term …

ESD.36 System Project Management
Fall 2012