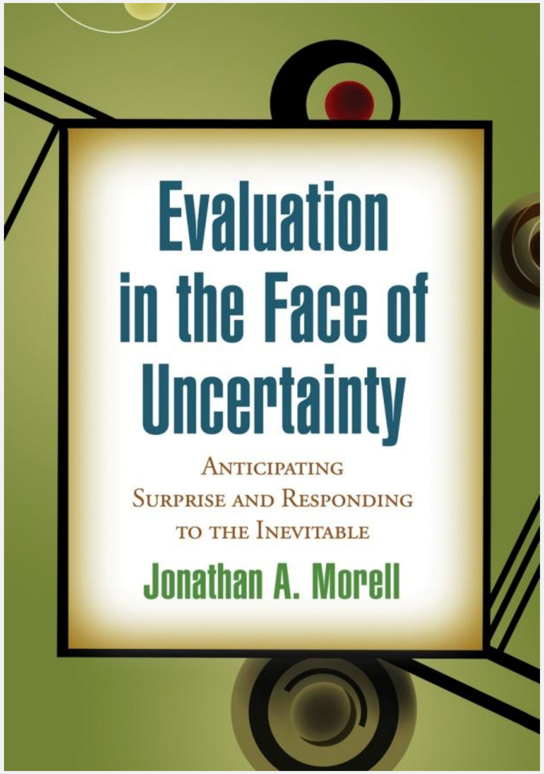
Evaluation in the Face of Uncertainty: Anticipating Surprise and Responding to the Inevitable

AEA/CDC Summer Evaluation Institute
June 13 – 16, 2010
Atlanta GA

Jonathan A. Morell, Ph.D.
jamorell@jamorell.com
(734) 646-8622



What is this workshop about?

Response to surprise

- Crisis response → advance planning

Disseminating knowledge

- Tactics for adding surprise to the evaluation mix

Community building

- More and better tactics
- More and better theory
- Archive of cases

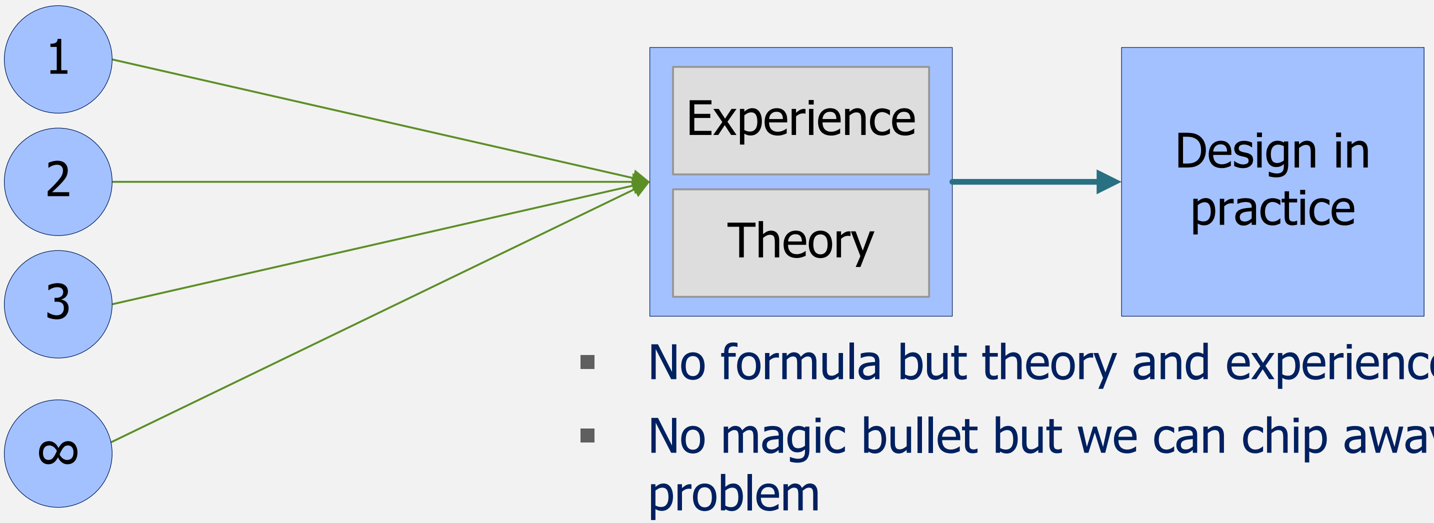
Adding "surprise" to evaluation planning

- Funding
- Deadlines
- Logic models
- Measurement
- Program theory
- Research design
- Information use plans
- Defining role of evaluator
- Logistics of implementation
- **Planning to anticipate and respond to surprise**

In this workshop we will go heavy on tricks and tips, light on theory, explanation, or analysis of collected cases.

The goal is informed commitment to practical action

- When is the likelihood of surprise high?
- When will surprise disrupt evaluation?
- If probability of disruption is high, what can we do about it?



- No formula but theory and experience help
- No magic bullet but we can chip away at the problem

- Many choices, one actual design
- All have pros and cons
- Tradeoffs are inescapable

Some historical background

We know why unexpected events occur

Evaluation

- Goal free evaluation emphasizes what a program does, not what it claims
- Interactivity between evaluation and the program being evaluated

Explanations embedded in domain

- Marketing, education, drinking regulation, tobacco control, product development, welfare, and many others, I have no doubt.

Complex systems

- Uncertain environments, cross linkages, self organization, adaptation, feedback loops with different latencies, etc.

But what to do about it as evaluators?

?

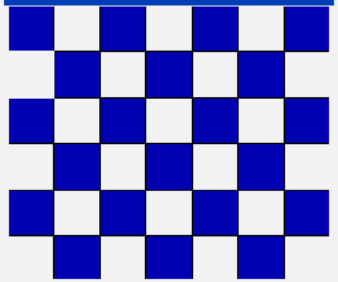
Guaranteed solution

- Post-test only
- Experimental group only
- Unstructured data collection

But we want to do a lot better

You can never tell the future but some surprises are more foreseeable than others

Foreseeable



- Get lucky
- Knowledge from stakeholders
- Good program theory
- Use research literature
- Use experts

Unforeseeable

- Complex system behavior makes prediction impossible no matter how clever we are.

PS – do not assume that complex systems are always unpredictable!

Theory

Limiting time frames

Exploiting past experience

Forecasting & program monitoring

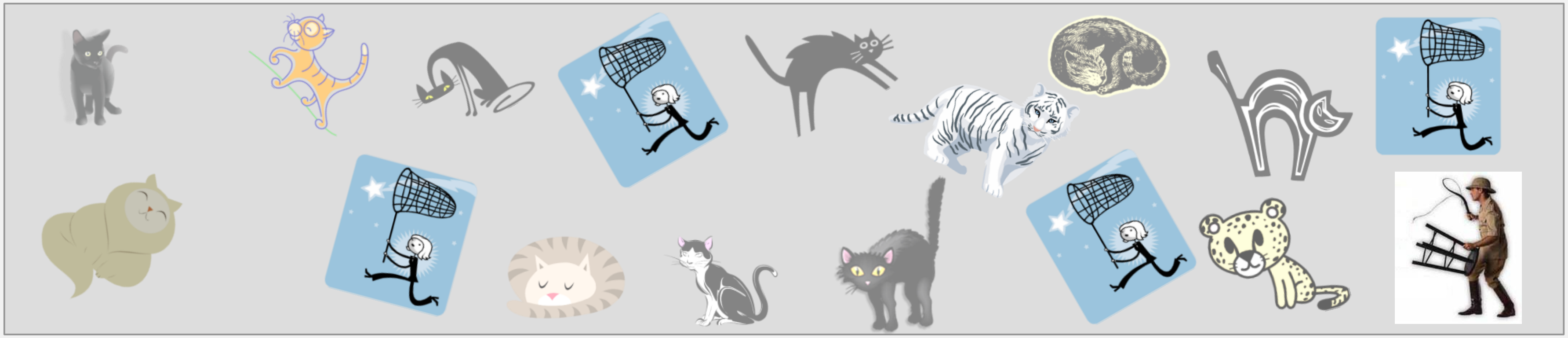
System based logic modeling

Retooling program theory

Agile methodology

Data choices

Jonny's favorite metaphor



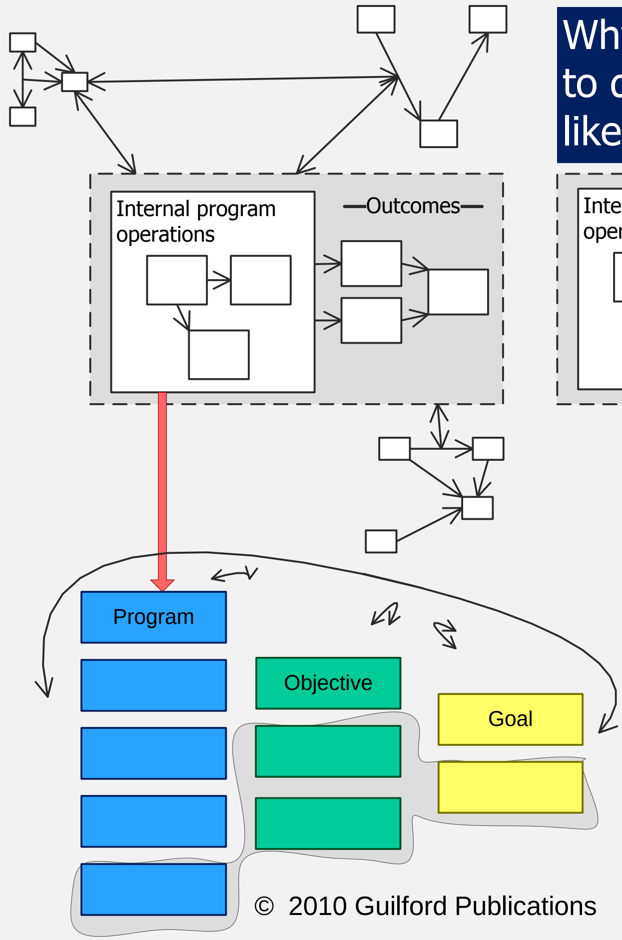
We don't know exactly where the cats are but we can sweep them toward one side of the landscape, and tame the one's that escape.

Programs and their evaluations have an essential similarity

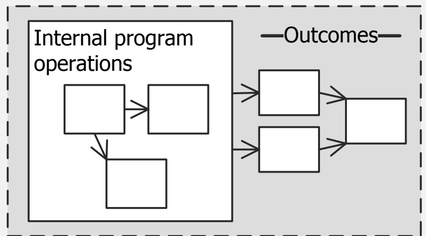
- What will help us with unexpected program outcomes will also
- Help us with unexpected problems in conducting an evaluation because
- Both are similar social constructions
  - Resources (time, people, $)
  - Processes
  - Embedded in a social setting
  - To accomplish specific objectives

What are the practical and political reasons for surprise?

If people are smart enough to know that the world looks like this



Why are they forced to design programs like this?



- Any single organization has limited money, political capital, human capital, authority and power
- Narrow windows of opportunity
- Competition requires bold claims
- Resource owners have parochial interests
- Design expertise limited
- Collaboration across agency boundaries is very difficult

- Short term success is rewarded
- Partial solutions can accrue to major success over time
- Pursuing limited success with limited resources is justifiable.

<u>Result</u>

- Narrow programs
- Simple program theories
- Small set of outcomes

Planners may know better but they are doing the best job they can. Evaluators have to follow.

What might an unforeseen but predictable outcome look like?

Program	Innovation	Results
▪ Post-natal care in Niger ▪ Formal fees ▪ Informal fees integrated into (hidden in) overall fee structure	▪ NGO provides drugs and supplies	Patients: drug hoarding (patients learned from previous programs)
	▪ Remove fees	Staff: game system, new fees

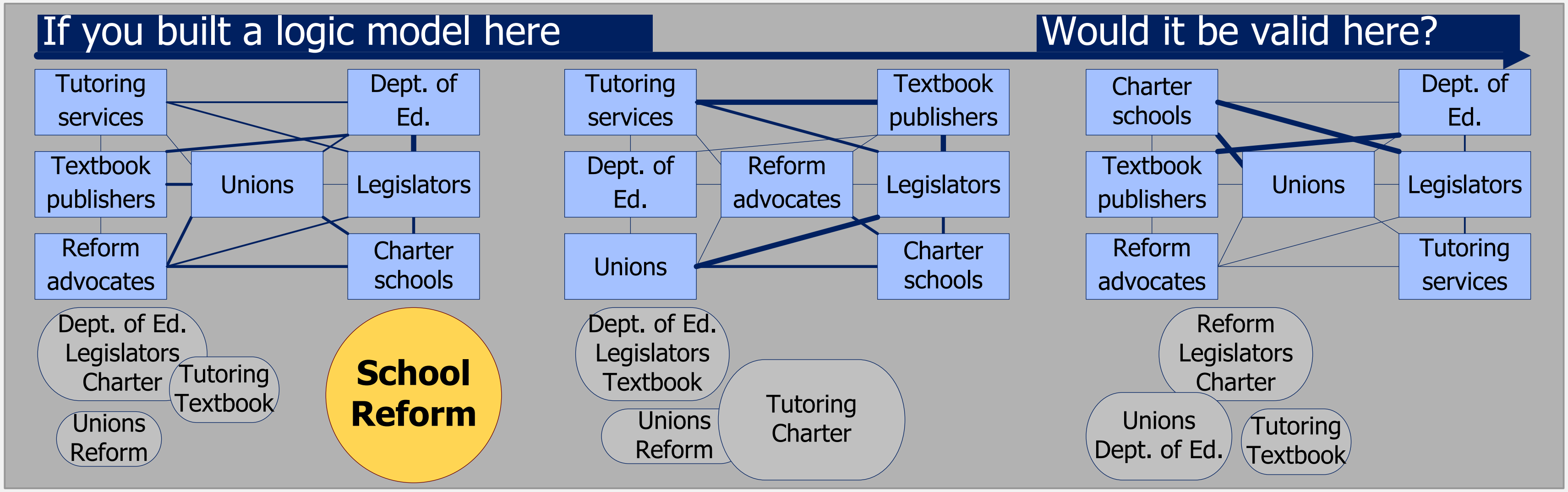
- Experience with similar programs
- Psychology of self interest
- Common sense

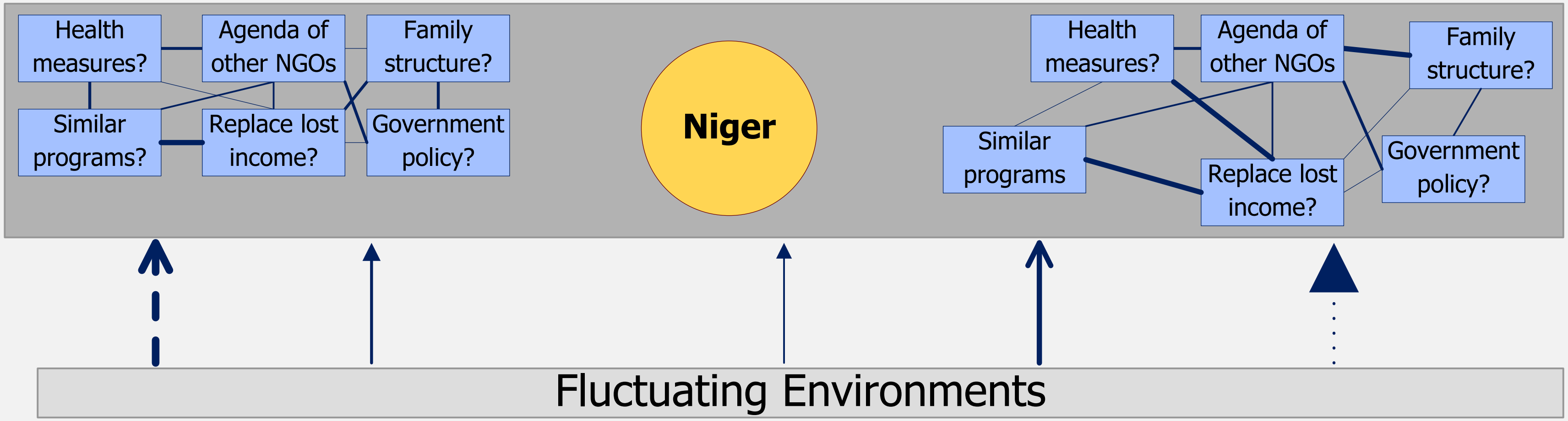
→

Something like this will happen, even if we can't say exactly what.

What might unforeseeable outcomes look like?

The problem is not sensitive to scale. We run into the same trouble with large and small problems.





How much surprise should we expect?

- Where is the program in its life cycle? Start-up phase is unstable.
- How stable is the environment? The past not be a good guide, but maybe better than nothing
- How robust has the innovation been over time and context?
- How rich and tight are the linkages?
- What is the "size" of the program relative to the boundaries of the system it is in?

	Size	
Linkages	High	Low
High	▪ Whole school reform ▪ Continuity of care	
Low		▪ New reading curriculum ▪ Pre-surgical checklist

<u>Why is this advice problematic</u>

- What does "big" mean with respect to a system and an innovation?
- What pattern of linkages qualify as "rich"?
- What feedback latency constitutes "tight"?
- What does "fast" mean with respect to a life cycle?
- Rich linkages might indicate both stability and fragility
- Small changes can have disproportionally large effects

But it still helps to ask the questions

How do we estimate the likelihood of surprise?

- Fidelity and robustness
  - Fidelity = extent program adheres to proven protocols
  - Robustness = program works when fidelity is low and context variation is high
  - Low fidelity + low robustness = high likelihood of surprise

- Time erodes predictability
  - Shifting environments
  - Longer feedback loops
  - Changing internal operations
  - New customer and stakeholder needs

- R&D content
  - Proven knowledge in novel setting, e.g. cross functional continuous process improvement in a poisonous labor/management climate
  - Novel program, e.g. injecting "consumer operator services" into a traditional mental health setting
  - Novel phenomenon, e.g. integrating Web 2.0 into routine organizational operations

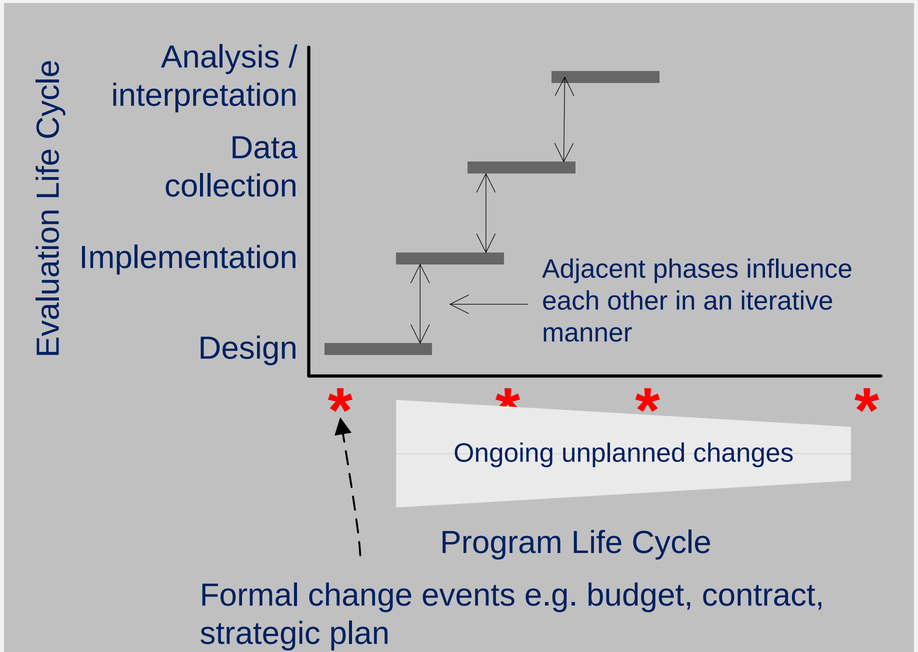
But when uncertainty is high AND uncertainty is problematic for evaluation?

- Life cycle view
- Social/organizational view

Program x evaluation life cycles can help us understand when uncertainty is high and problematic

Evaluation life cycle

- Shorter or longer than program life cycle
- Begins sometime after program start (usually)
- Stages affect each other iteratively
- More spiral than waterfall form, but with some lag, all stages are present



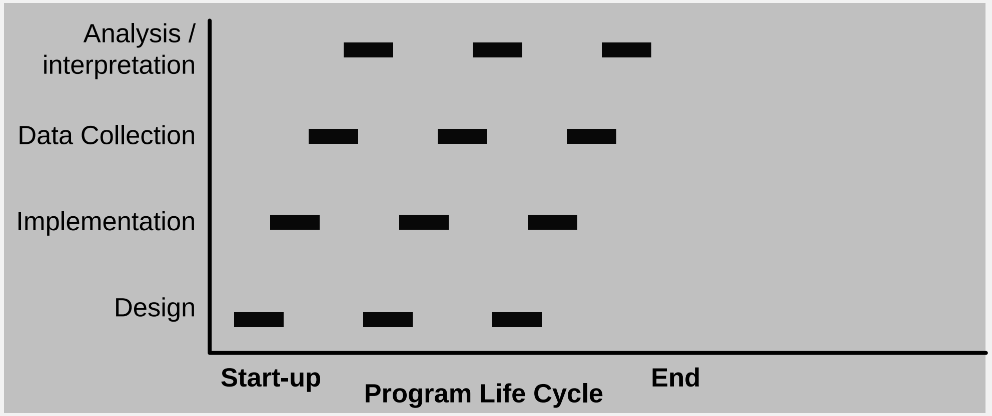
Program life cycle

- Formal events, e.g. budgets, yearly plans, publishing RFPs
- Continual stream of micro-level changes and environmental adaptations with greater effects early on

Relationships between the life cycles affect unpleasant surprise

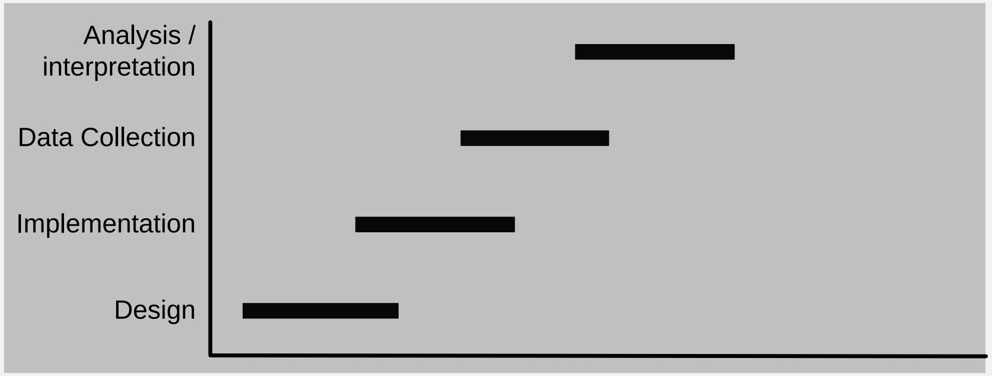
Multiple, short term studies

- Continuous process improvement
- Short time between cause and effect = inference with simpler methodology
- Pretesting and prototyping to test evaluation design
- Inherently sensitive to unexpected program activity



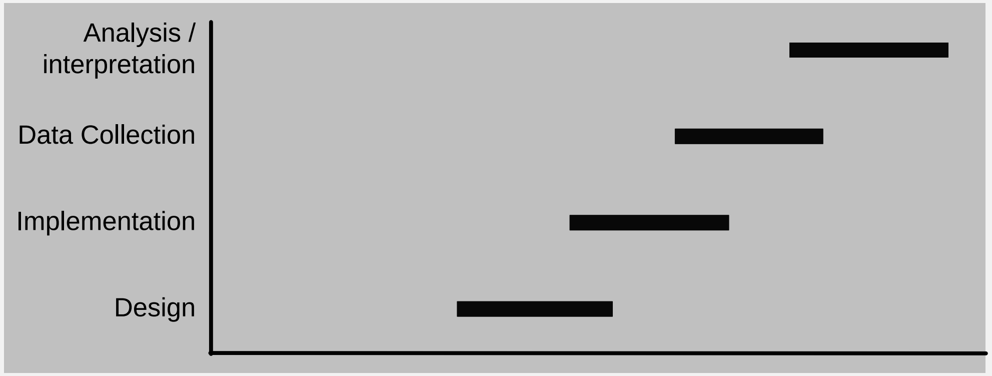
1:1 Correspondence between life cycles

- Fog of start up
- Surprise late in program life cycle can force early stage evaluation redesign
- *Gets worse when design and data requirements must be stable over time*



Retrospective focus

- Emphasis on program in stable part of life cycle
- Program change, evolution relatively unimportant



Stage where corrective action most useful	Stage where surprise discovered: Design	Implementation	Data Collection	Data Analysis
Design	Case 2			Case 1
Implementation				
Data Collection				
Data Analysis				

Case 2: Computer training

- Early discovery of disagreement over multiple stakeholders' priorities
- Design reworked many times prior to evaluation implementation
- Design was able to satisfy all needs

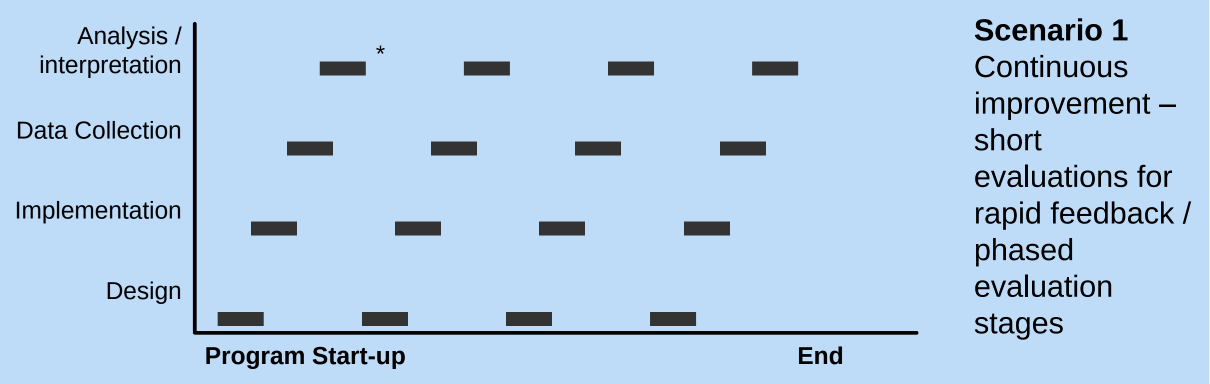
Case 1: Child Care

- Sponsor's priority: ratio of caregivers to children
- Minimum ratio set by regulation
- Upper limit set by economics
- Restricted range → no significant findings
- Design problem discovered at analysis stage
- Evaluation question morphed: Impact of number of children per group.

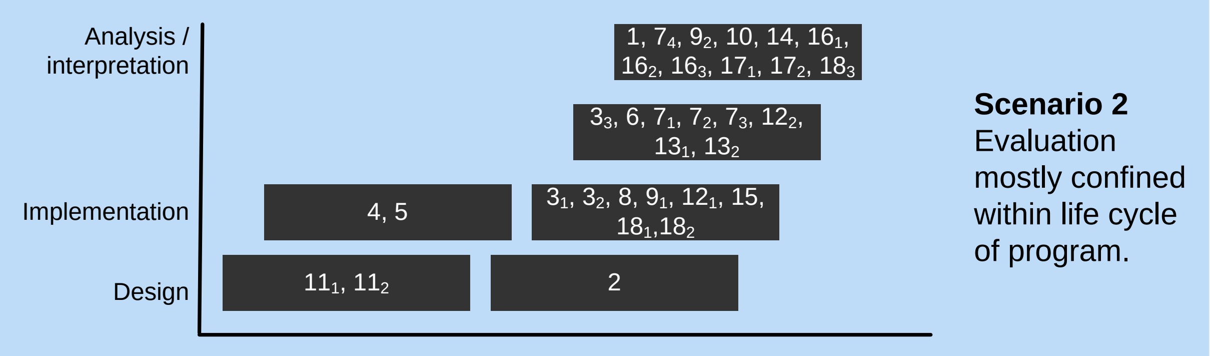
We can also learn a lot by comparing evaluation stage where a problem is discovered to the stage where it is best fixed.

Where does surprise fall on the program x evaluation life cycles? 32 surprises from 18 cases

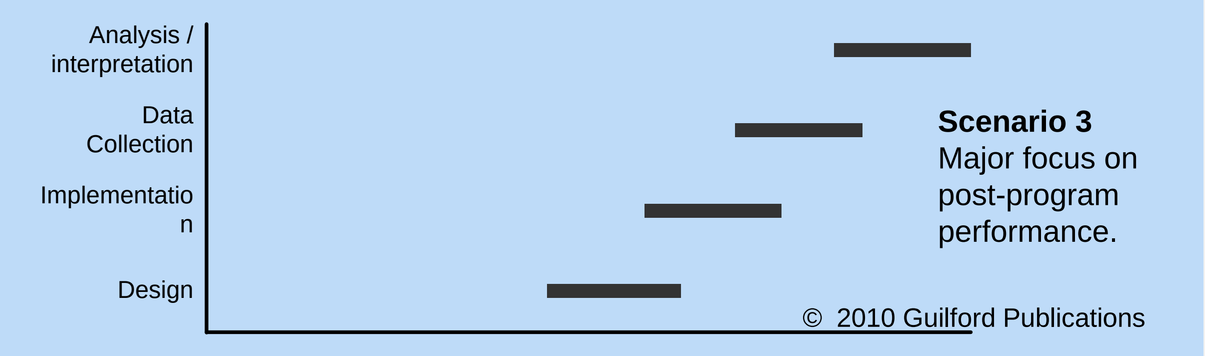
This is a CI-like approach. I wish this kind of evaluation were done.*



The common work we all know and love



- Not enough of this kind of work done?
- I could not find it?
- Less susceptible to surprise?



* Morell, J.A (2000) Internal evaluation: A synthesis of traditional methods and industrial engineering . *Am. J. of Evaluation*

Where does surprise come from? A Social/organizational view is also helpful in understanding surprise

Program related

Environment
- Funder / regulatory decisions
- Program client / customer behavior

Internal activity
- Organizational behavior
- Program staff

Evaluation implementation

- access to interviewees
- IT system capabilities
- access to comparison groups
- innovation creep across groups
- ...

Adjustments to evaluation

Design
e.g.
- substitute time series for cross-group measures
- add interviews to explicate program theory

Procedure, logistics
e.g.
- Change method of recruiting respondents
- Hire professional interviewers

Analysis / interpretation
e.g.
- construct proxy variables
- redefine study as combined impact of control and experimental groups

Where does surprise come from and how does it move through the system? 32 surprises from 18 cases

Program related

Environment

- Funder / regulatory decisions
- Program's client / customer behavior

1, 2, 6, 9_1, 11_1, 11_2, 12_2, 13_1, 13_2, 14, 16_1, 16_3*, 18_1, 18_2,

Internal activity

- Organizational behavior
- Program staff

3_1, 3_2, 3_3, 4, 5, 7_1, 7_2, 7_3, 7_4, 8, 9_2, 10, 12_1, 15, 16_2, 16_3*, 17_1, 17_2, 18_3,

Evaluation implementation

- Program – evaluation coordination
- access to interviewees
- IT system capabilities
- access to comparison groups
- innovation creep across groups
- ...

3_1, 3_2, 3_3, 4, 5, 6, 7_1, 7_2, 7_3, 8, 9_1, 11_1, 11_2, 12_1, 12_2, 13_1, 13_2, 15, 18_1, 18_2

Adjustments to evaluation

Design

e.g.

- substitute time series for cross-group measures
- add interviews to explicate program theory
- New evaluation objectives

2, 3_1, 3_2, 8, 10, 15, 18_1,

Procedure / logistics

e.g.

- Change method of recruiting respondents
- Hire professional interviewers

4, 7_3, 9_1, 12_1, 12_2, 13_1, 13_2, 14

Analysis / interpretation

e.g.

- construct proxy variables
- redefine study as combined impact of control and experimental groups

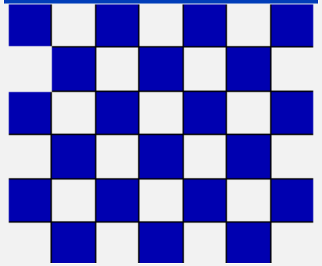
1, 3_3, 5, 6, 7_1, 7_2, 7_4, 9_2, 11_1, 11_2, 16_1, 16_2, 16_3, 17_1, 17_2, 18_2, 18_3

Some other useful ways of categorizing sources of surprise

- Pilot tests / feasibility assessments: Important but not infallible, e.g.
  - Last year's data used to estimate power do not apply to current year
  - Query individuals who can answer for themselves but not for the organizational behavior
- Resistance to evaluation
  - People think they can speak for a program when they can't
  - Levels are not static over time
- Incorrect assumptions early in the evaluation life cycle
  - Funders ask the wrong question
  - People think they can promise data but can't deliver

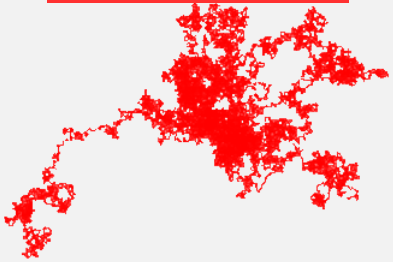
These methods are most useful early in evaluation life cycle

Foreseeable



- Get lucky
- Knowledge from stakeholders
- Good program theory
- Use research literature
- Use experts

Unforeseeable



- Complex system behavior makes prediction impossible no matter how clever we are.

PS – do not assume that complex systems are always unpredictable!

Theory

Limiting time frames

Exploiting past experience

Theory as a tactic for reducing surprise

▪ Why is theory useful? ▪ Example 1: Program theory* ▪ Example 2: Life cycle behavior ▪ Example 3: Perfect Market	Explanatory power helps look in the right place.
▪ Why is theory problematic?	Too many to choose from
▪ How can value be maximized and problems minimized?	Choose more than one, choose wisely.

* For much more on program theory and logic models see workshop slides: Logic Models: Uses, Limitations, Links to Methodology and Data American Evaluation Association Annual Meeting – Orlando FL November 10th, 2009. Downloadable from www.jamorell.com

Program Theory

- Context specific
- Engages stakeholders
- Good framework for surfacing assumptions
- Captures knowledge of deep program experts
- Assures evaluation that will meet what stakeholders perceive as their needs

- Stakeholders cherished beliefs can be wrong
  - Limited to stakeholders' perspectives
  - Not likely to capture much relevant knowledge
    - Similar programs in other contexts
    - Research literature

Theory examples: Life cycles

Ex #1 Worker participation safety program

> Union member to evaluator: "These things last 5 years. They always do."

Ex #2 CC Training

				Student Placement Y 1	Student Placement Y 2	Student Placement Y 3	Technology refresh rate
Welding	Verify job market →	Instruction quality →	Student learning →	High	High	High	60 months
Animation	Verify job market →	Instruction quality →	Student learning →	High	High	Low	18 months

Theory example: Perfect market in health service choice

Recognizing that measurement and public reporting are powerful mechanisms to drive quality and efficiency improvement throughout the health care system, purchasers and consumers have embraced a vision of a transparent health care market, in which decision-making is supported by publicly reported comparative information. Our shared vision is that with this information, Americans will be better able to select hospitals, physicians, and treatments based on nationally standardized measures for clinical quality, consumer experience, equity, and efficiency. http://www.healthcaredisclosure.org/about/

Choosing Theories

Principles

- One is better than none
- A few are better than one
- Include stakeholders' program theory
- Using more than a few is dysfunctional – too many variables and relationships
- Choices establish path dependency. Make sure all theories in pool are relevant

Thought Experiment

1. Stakeholders establish program theory
2. Recruit group of diverse experts
3. Experts choose 5 other *relevant* theories
4. Pick 1/5 at random
5. Add to stakeholder program theory
6. Develop evaluation
7. Pick another theory
8. Repeat

Result: Similarity across designs

- Same program
- Same stakeholders
- Same environment
- Same information needs

Result: All designs better than if only 1 used

- Stakeholders provide context specificity
- Other theories provide relevant
  - Variables
  - Relationships

Capitalizing on what we already know

Few programs are so unique that previous experience won't decrease surprise

- Process knowledge: What happens to programs like mine in similar circumstances?
  - E.g. How do needle exchange and health eating programs fare at election time?
- What do we know about how programs like mine work?
  - E.g. Do threatening public service announcements encourage diabetics to monitor their blood sugar and control what they eat?
  - Literature reviews and interviews work

Example of using process knowledge to understand program behavior

	Certainty of outcomes	Political Sensitivity
Use of pre-surgical checklists	High	Low
Dissemination of evidence based practice data	Low	High

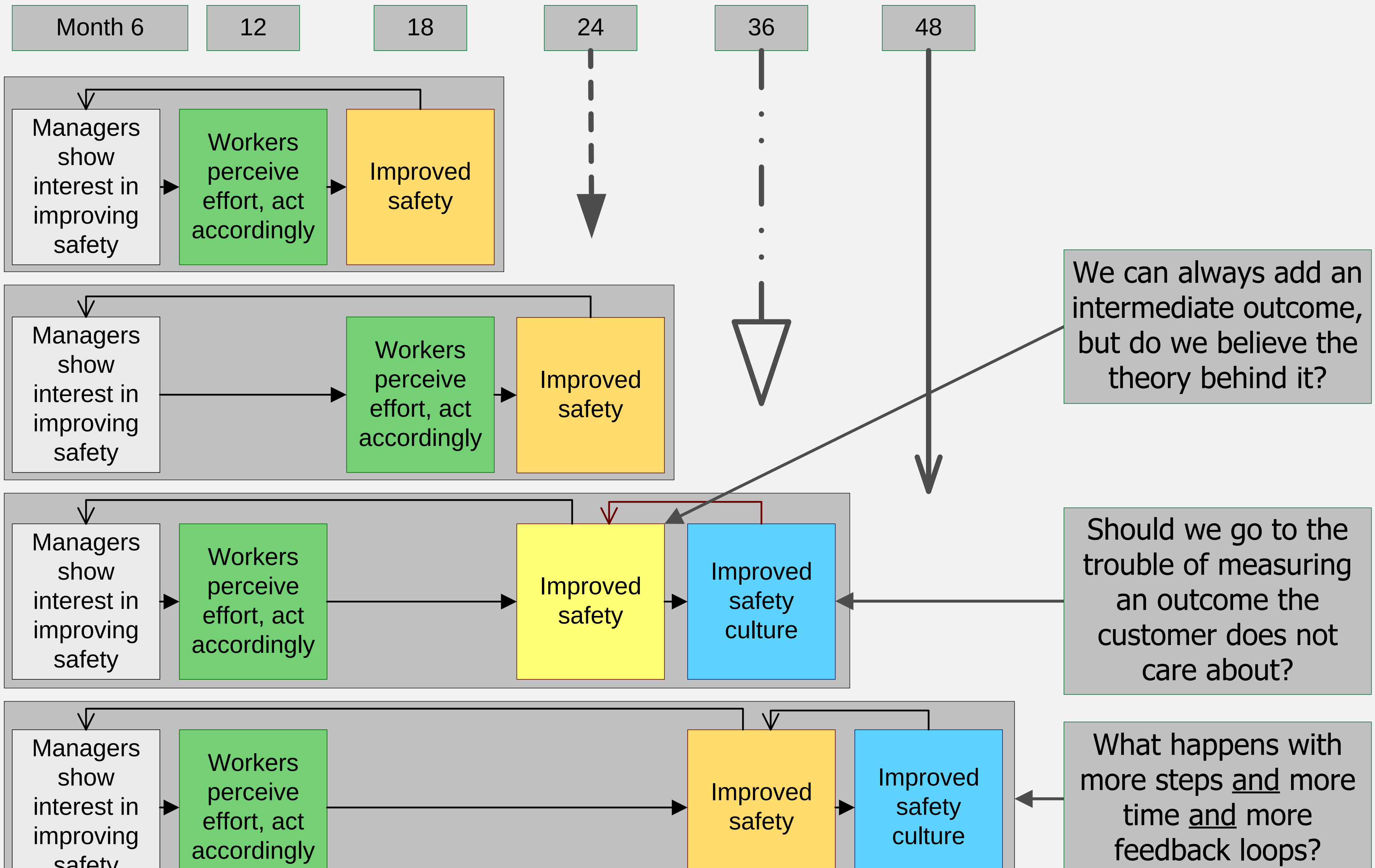
Example of using domain knowledge

- Tobacco control: Integrate person focus and environmental focus
- Problem: Not enough known about successful implementation in this context
- Solution: 1) Literature review of successful ecological implementations. 2) Theory based evaluation of application for tobacco control

Choosing knowledge domains: Principles are the same as with theory

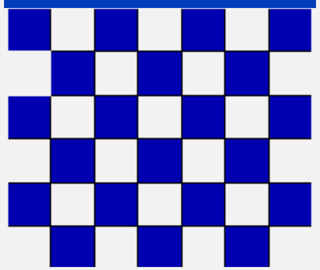
- One is better than none
- A few are better than one
- Include stakeholders' program expertise
- Using more than a few is dysfunctional – too many variables and relationships
- Choices establish path dependency. Make sure all candidates are relevant

We can minimize surprise by limiting temporal and causal distance, but we better be careful. A lot can happen as time marches on.

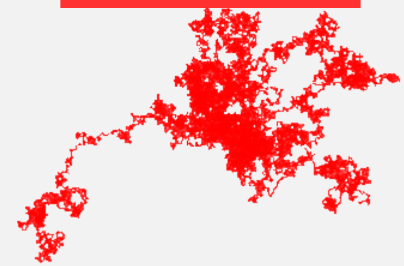


These methods are most useful for detecting leading indicators

Foreseeable



Unforeseeable



- Get lucky
- Knowledge from stakeholders
- Good program theory
- Use research literature
- Use experts

- Complex system behavior makes prediction impossible no matter how clever we are.

PS – do not assume that complex systems are always unpredictable!

Forecasting & program monitoring

System based logic modeling

The trick is to do a little better than the Delphic oracle

Use planning and monitoring techniques to revisit program *and* evaluation at various slices of their life cycles

- Assumptions underlying program success
  - Which are critical?
  - How robust or brittle?
  - Indicators of failure?
- Future states
  - What is the desired future?
  - What are the likely futures?
- Environmental conditions
  - Funding / Politics / Culture
  - Needs of service population, whether individuals of organizations
- Internal operations
  - Staff makeup, organizational structure/culture

How to get all this information?

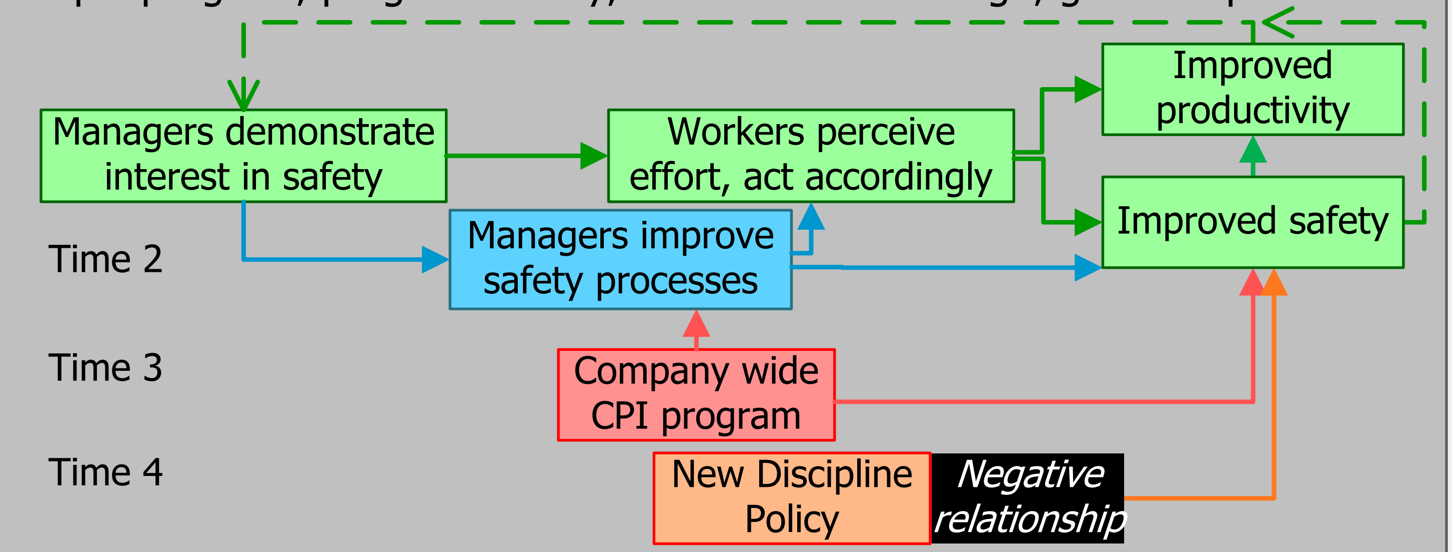
- Stakeholders are necessary but not sufficient
- Identify all relevant domains
- Identify most relevant subset
- Query relevant subset frequently
- Rotate thorough the others
- Use case study methods

Example of how a program may change over time

<u>The program: Improve safety by training managers</u>

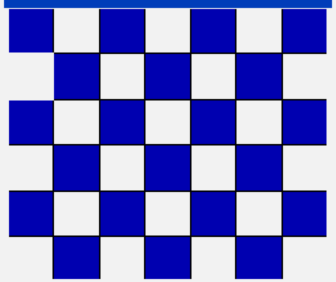
- Some program assumptions
  - Workers can interpret managers' behavior
  - Safety → productivity
  - Safety + productivity → manager behavior
  - *No linkage with other CPI initiatives*
  - *No activity to sabotage program*
- Some evaluation assumptions
  - Need only manager, worker surveys + safety, productivity data
  - No confounds to causal inference

Simple program, program theory, and evaluation design, gets complicated.



Agile Evaluation

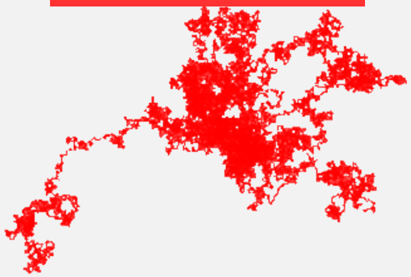
Foreseeable



- Get lucky
- Knowledge from stakeholders
- Good program theory
- Use research literature
- Use experts

How can an evaluation be designed to change?

Unforeseeable



- Complex system behavior makes prediction impossible no matter how clever we are.

PS – do not assume that complex systems are always unpredictable!

Data choices

Agile methodology

Retooling program theory

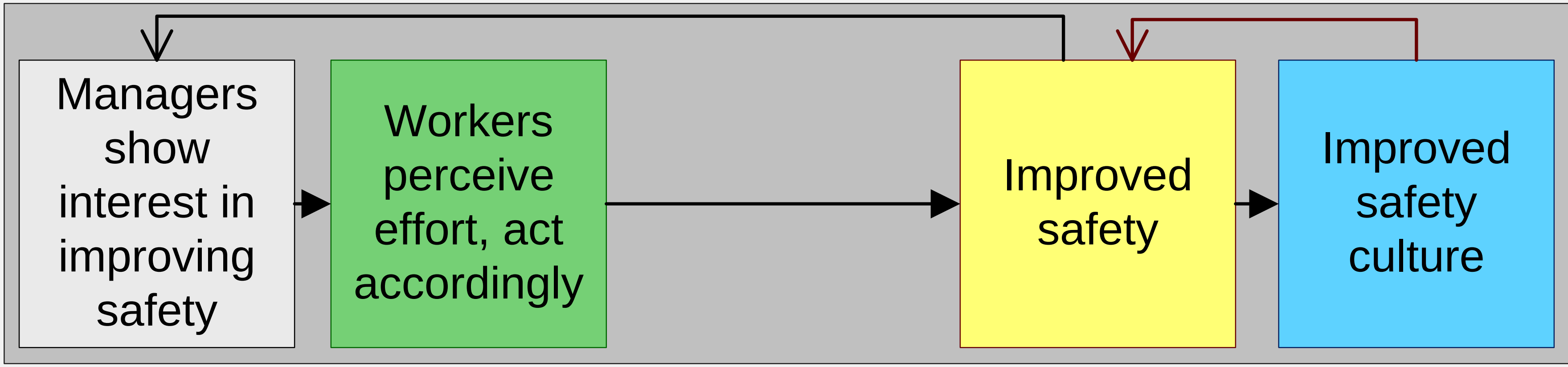
Data

Can the data be modified to meet new needs?	e.g. ▪ Validated scales vs. open ended questions ▪ Custom programming vs. standard lookup ▪ Structured teacher observations during class vs. casual assessment by visitors
Is gatekeeper approval needed?	e.g. ▪ OMB ▪ Air Force Survey Office ▪ Corporate VP
Are substitutes available without harming the intent of the evaluation?	e.g. ▪ Self report → clinical record ▪ Direct cost → total cost
Are substitutes practical?	▪ Collection burden increase ▪ Development cost to move to new methods ▪ Switching time relative to deadline for getting data ▪ E.g. Clinical records vs. patient report

Agile methodology: Some definitions

Agile	Ability to change quickly.
Methodology	Logic in which observations are embedded
Evaluation	Organizational entity • Processes • Resources • Structures Constructed to allow • Data acquisition to feed • Methodology that allows data interpretation
How to make an evaluation agile	• Flexible vs. rigid design elements • Dependencies • Boundaries • Partition • Retool program theory

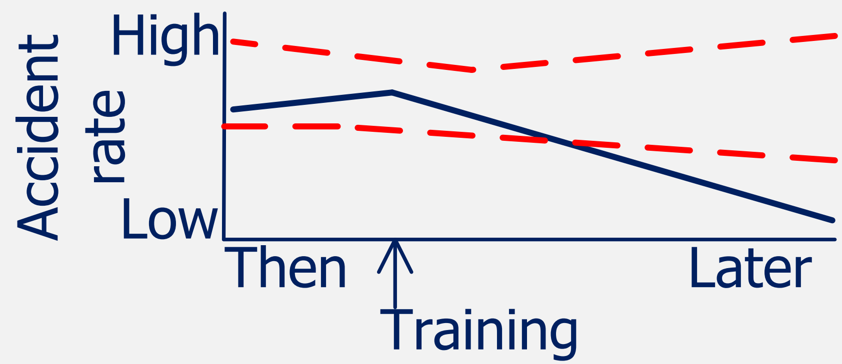
Example of agile and brittle evaluation components



Methodology

- Two possible comparison groups
- Time series and cross sectional possibilities
- If any one comparison goes away others remain

Experimental Control Time Series



Data

- Develop, validate fixed-choice instruments for pre-post training assessments
- Interviews ½ way through training for course improvement

Data

- Interviews with workers soon after an accident to see why/if manager behavior affects safety
- Safety, accident, derailment statistics from IT systems to test primary outcome

What are the agile and brittle components?

Data: Formative	Data: Summative	Design	Implications for Agility
Validated instrument test training quality		2, beginning, end of training	▪ Time, cost: difficult to change instrument ▪ Timing to training critical
Semi-structured questions: if/why managers change		1 half way through for course improvement.	▪ Minimal effort to determine questions. ▪ Variation around midpoint OK.
	Validated safety culture scales	3, start, end, 6 months post	▪ Time, $, difficult to change instrument. ▪ First 2 timed to training. 3rd can move
	Interviews: why manager behavior affects safety	Keyed to occurrence of accidents.	▪ Minimal time to determine questions. ▪ Synchronize with accidents
	Safety & accident stats	From company IT system	▪ Available any time ▪ Not linked to training
		1- Control groups other parts of company	▪ Difficult to implement. Considerable negotiation needed.
		2- Time series on accidents	▪ Available from IT systems. Fallback if #1 disappears

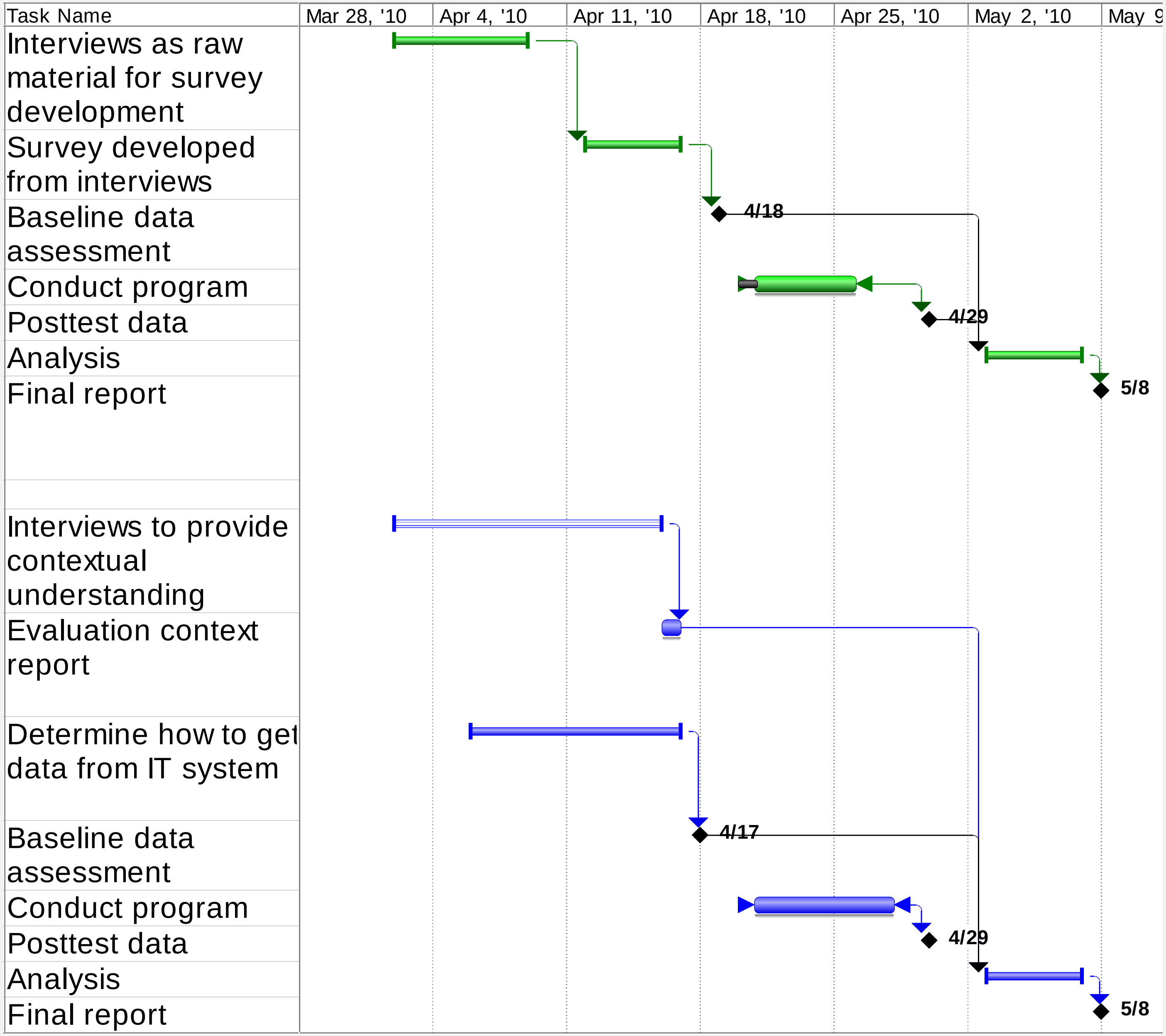
Number and Richness of Dependencies Affect Agility

- Which design serves the customer best?
- Which design is riskier?
- Which should be chosen?

Design 1

- Same program
- Somewhat different evaluation questions
- Different in length of critical path

Design 2



Example 1: Evaluations that depend on managing boundaries are not agile

Example #1 Negotiating for access to data			
Organizational distance	**Data collection burden**	**Data sensitivity**	**Agility**
Different leaders	Interviews	Labor / management interactions	Low. Renegotiating any evaluation condition is difficult.
Same leader	IT data	Technical capacity	

There are many good reasons to choose one or another design. Agility can be <u>one</u> of them.

Example 2: Evaluations that depend on managing boundaries are not agile

Example #2 Control Groups	
	Agility
Design based on random assignment	Low. Even small breakdown can have large impact on analysis
Design based on naturally occurring groups	

There are many good reasons to choose one or another design. Agility can be <u>one </u>of them.

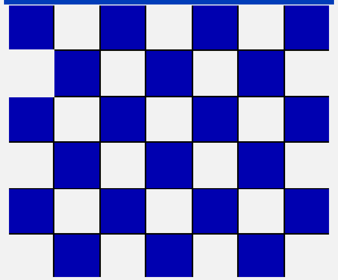
One reason for partition in an evaluation design is agility

We always split our projects into phases in the service of practicality, methodology, and promoting knowledge use. E.g. pretests of instruments, pilot studies to estimate power, preliminary findings to test stakeholder needs, feasibility assessments

Agility can be another reason to think about partition			
	Original	**Innovation for Agility**	**Advantages and Disadvantages**
Process example	Interview 1/2 way in training. Feedback on instruction	Interview some ¼, ½ and ¾ through	▪ More opportunity to see if program is working to plan. Chance to change outcome measures ▪ Logistics more difficult ▪ Opportunity for as much information as possible at ½ point is lost
Outcome example	Download IT data at end	Analyze at intervals	▪ Chance to see detect unexpected outcomes ▪ More evaluation resources for analysis ▪ Greater burden on company’s IT staff

Selecting tactics: Sometimes more is not better

Foreseeable



Unforeseeable

- Get lucky
- Knowledge from stakeholders
- Good program theory
- Use research literature
- Use experts

- Complex system behavior makes prediction impossible no matter how clever we are.

PS – do not assume that complex systems are always unpredictable!

Theory

Limiting time frames

Exploiting past experience

Any few of these may make sense.

Forecasting & program monitoring

System based logic modeling

Retooling program theory

Agile methodology

Data choices

But all together they can get us into a lot of trouble.

Example of how multiple tactics induce new problems: Buffering against promised interviews not materializing

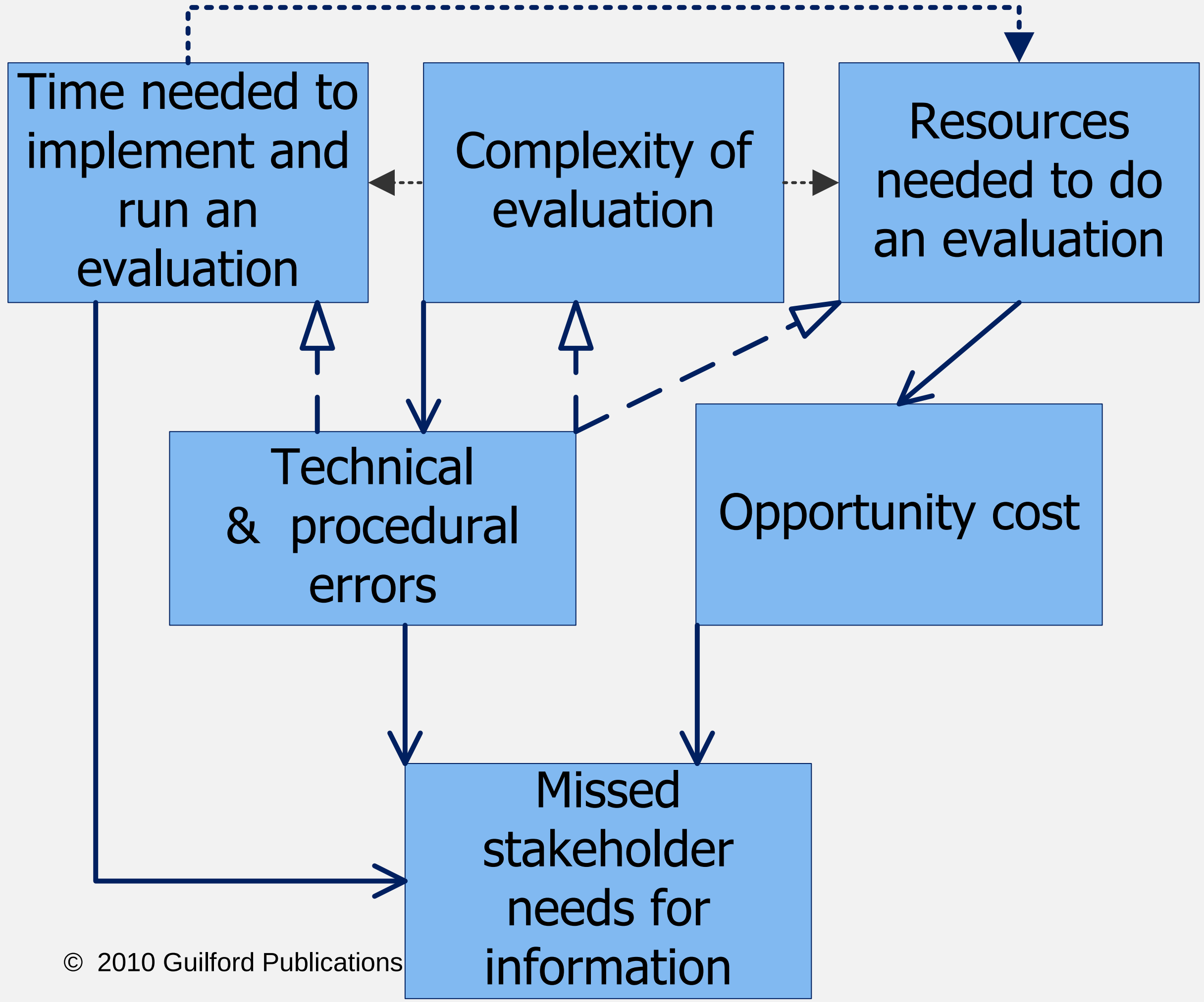
Evaluation Scenario	Advantages	Disadvantages
▪ Treatment ▪ 6 month follow-up, phone interviews by clinic staff ▪ 12 month, as above ▪ Administrator assures cooperation	Detailed information	Resistance to work not seen as serving clinical purpose
Eliminate 1 data collection	Might get needed information	▪ Less data ▪ Still no guarantee of cooperation
Eliminate interviews, rely on IT information instead	No clinical cooperation needed	▪ Sparse data ▪ IT data often untrustworthy ▪ A lot of work to vet systems
Do both	▪ Redundancy ▪ Increased range of information ▪ Multiple measures	▪ Longer to design and implement ▪ Need more diverse expertise on evaluation team ▪ Hard to maintain integrity of evaluation over time ▪ Nurture good relationships with clinical <u>and</u> IT staff ▪ Resources diverted, e.g. from analysis

Framework for appreciating trade-offs

Any change to time, complexity or resources can affect the others

Fixing technical or procedural problems can affect time, complexity or resources

Delay exceeds stakeholder need for information, but
Not fixing the problem also risks not meeting stakeholders’ needs



How can we design maximum protection against surprise before problems set in?

Tactics for dealing with surprise

- All tactics for dealing with surprise have social and technical aspects

Social

Technical

Groups

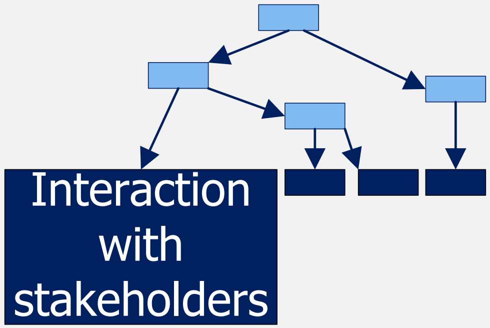
- Ability to detect and adapt rises with # of people and groups to provide advice
- But as size increases so do problems of management and decision making

Interaction with stakeholders

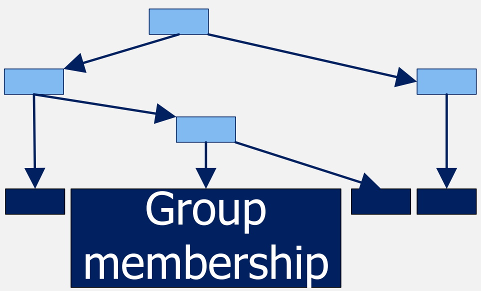
Group membership

Group behavior

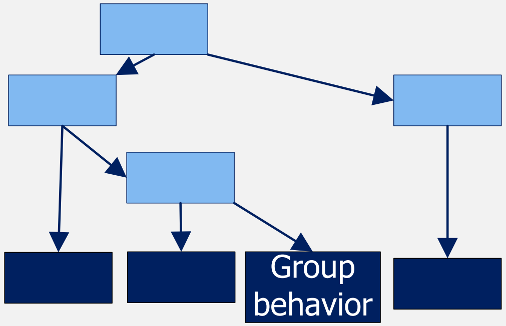
Design component connectedness



- General good practice to engineer as much communication as possible along the evaluation life cycle
  - New or evolving needs
  - Evaluation findings
  - Insight on analysis
  - Redesign logic models
- *Also sets a context where minimal extra effort or complication needed to discuss unintended or unexpected program or evaluation behavior*

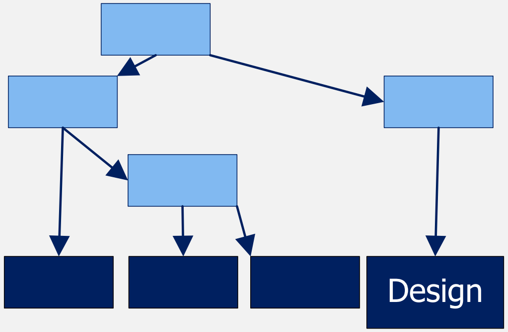


- Split essential and non-essential members
- Essential: stakeholders whose continued involvement is needed to
  - maintain the evaluation
  - make use of findings
- Non-essential: weak claims on the program but advice can be useful.
  - Not on the critical path
  - Relatively low cost
  - Very many possible groups but
    - Some are better than none
    - Membership can rotate over time



Diverse input means larger groups. Larger groups are hard to manage

- Use special techniques to get small group behavior from large groups
- Delphi methods to avoid discord
- Loose groups, e.g. advisory boards meet just frequently enough to know the project and who can provide occasional useful advice
- It's frequency, not just cost. Phone and Web conferencing lowers cost and increases amount of advice that can be purchased
- Split groups by recognize relative connectedness, e.g. sustainability and impact are related, but different enough to keep advisors separate.



- Evaluation plans differ in the number of critical paths among their components Make this ***one*** of the considerations. E.g.
  - 6 month follow-up data to design 12 month follow-up, ***or***
  - Design instruments based on cross sectional analysis of past service recipients at 6 and 12 months
- Richness of dependencies. E.g.
  - Continual iteration: 1) Simulation to determine program performance + 2) empirical data collection ***or***
  - Simulation after data collection