Building Knowledge Through Empirical Studies:
Generating and Building Support for Hypotheses

Jeff Carver
Experimental Software Engineering Group
University of Maryland

Research Goal

Study the effect of context variables, focusing on the variations in individual inspectors, in terms of their impact on defect detection effectiveness during a software inspection

Software Inspections

- Common process used for defect detection
  - Various methods (e.g. formal inspection)
  - Various techniques (e.g. PBR, OORTs)
- Most research focused on the method or technique used
- To answer deeper questions, research needs to focus on the context of the inspection
  - E.g. Time constraints, perceived value in organization, physical setting, human inspectors
- Little study has been done on the human inspector
  - Questions we would like to be able to answer
    - What type of knowledge is beneficial for an inspector?
    - What type of technique will be most useful in a given setting?

Qualitative Studies

- Problem: Difficult to answer complex SE questions with a purely quantitative approach because
  - Working with human subjects
  - Typically have small sample sizes
  - Experiments are expensive to run
  - Need some support for a hypothesis before investing effort in full experiment
- Solution: Use a qualitative approach that includes a quantitative aspect

Methodology

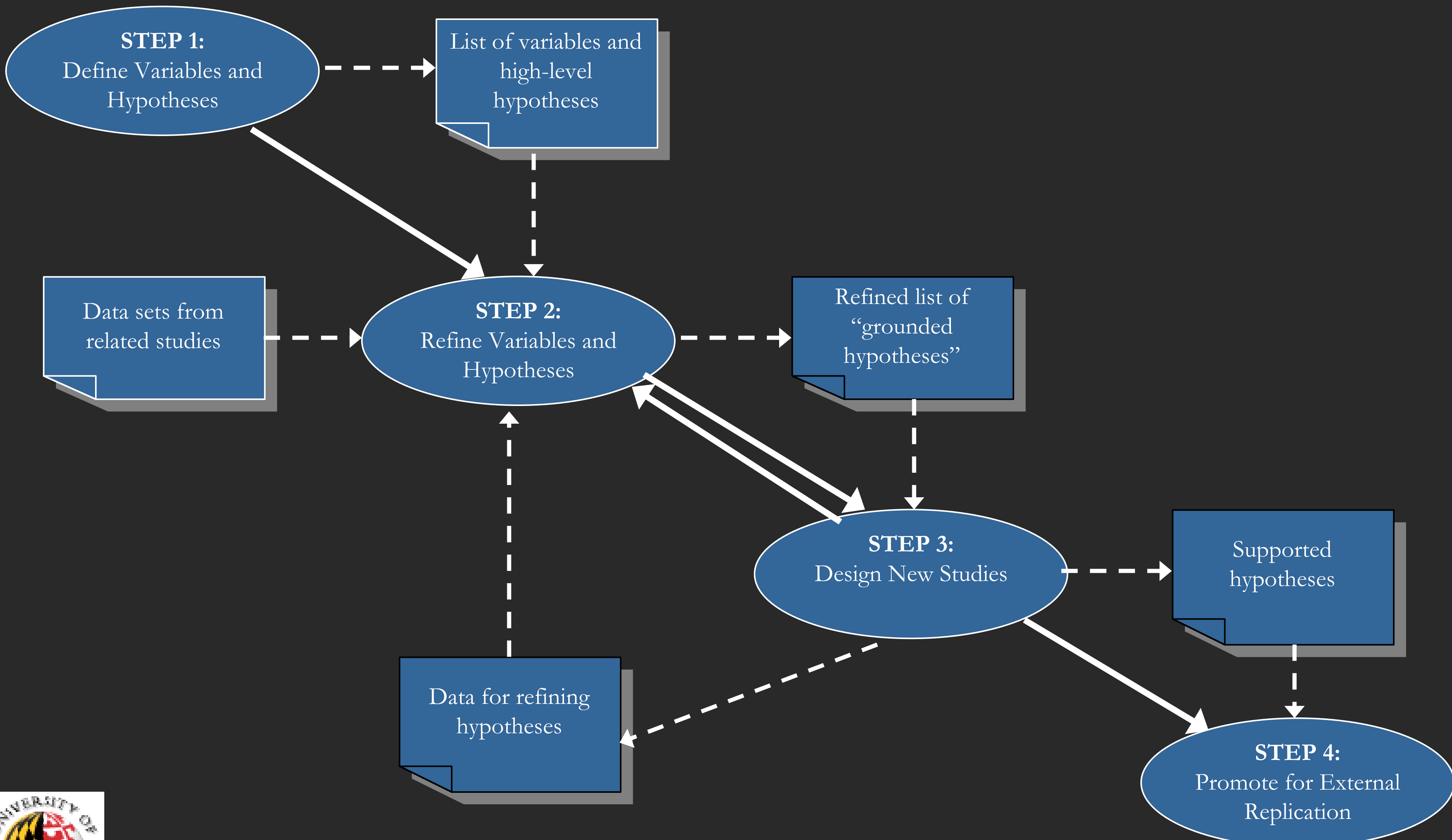
- Grounded Theory
  - Definition
    - Theory formed bottom-up from the data
    - Approach taken from Social Sciences
  - Constant Comparison Technique
    - Form theories (hypotheses) that accurately describe the data from the first set of data
    - For each additional data set, refine those theories (hypotheses) to take into account the new data
    - Continue until the theories (hypotheses) stabilize (have little change) or no more data sets are available

Methodology

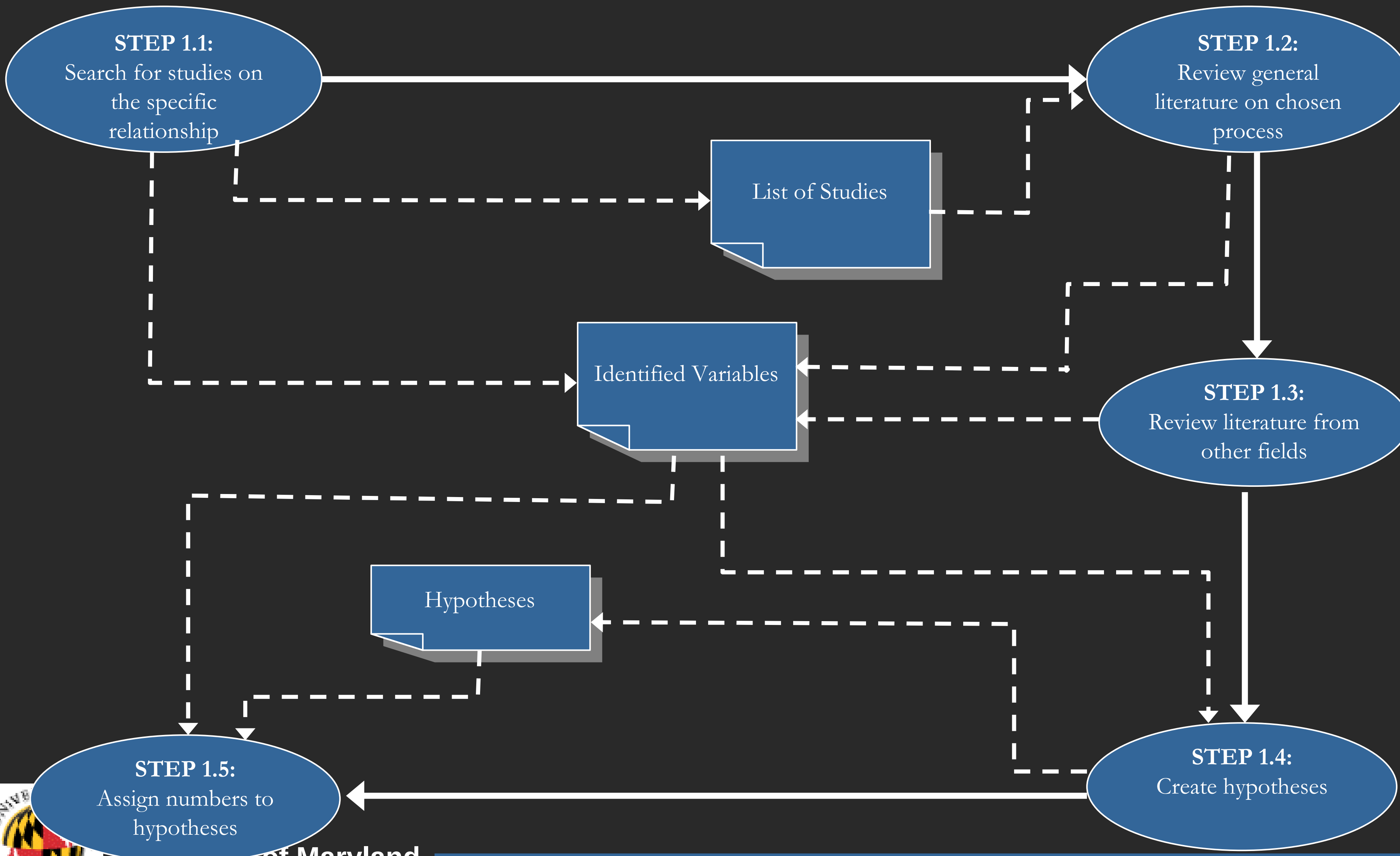
- Tailoring of Constant Comparison
  - Consult experts to identify potential variables and high-level hypotheses
  - Analyze existing empirical data to refine and “ground” hypotheses
  - Conduct new studies to test “grounded” hypotheses

- Overall Goal
  - Generate well grounded hypotheses to maximize the benefits of experiments designed to test those hypotheses

Methodology



Step 1: Define Variables



Step 1.1 – 1.3: Identify Variables

- Using studies published in the literature
  - Application Domain Knowledge (DK)
    - Dow94 at NASA; Fowler86 at AT&T; Parnas85 (Active Design Review)
  - Software Development Experience (DE)
    - Schneider92 (N-Fold Inspection); Cheng96 at Univ. New South Wales
  - Inspection Process Experience (PE)
    - Fusario97 IESE; Melo01 at Oracle
  - Training (T)
    - Collins89 and Pintrich96 from Education Literature
  - Motivation (M)
    - Alexander00 from Education Literature

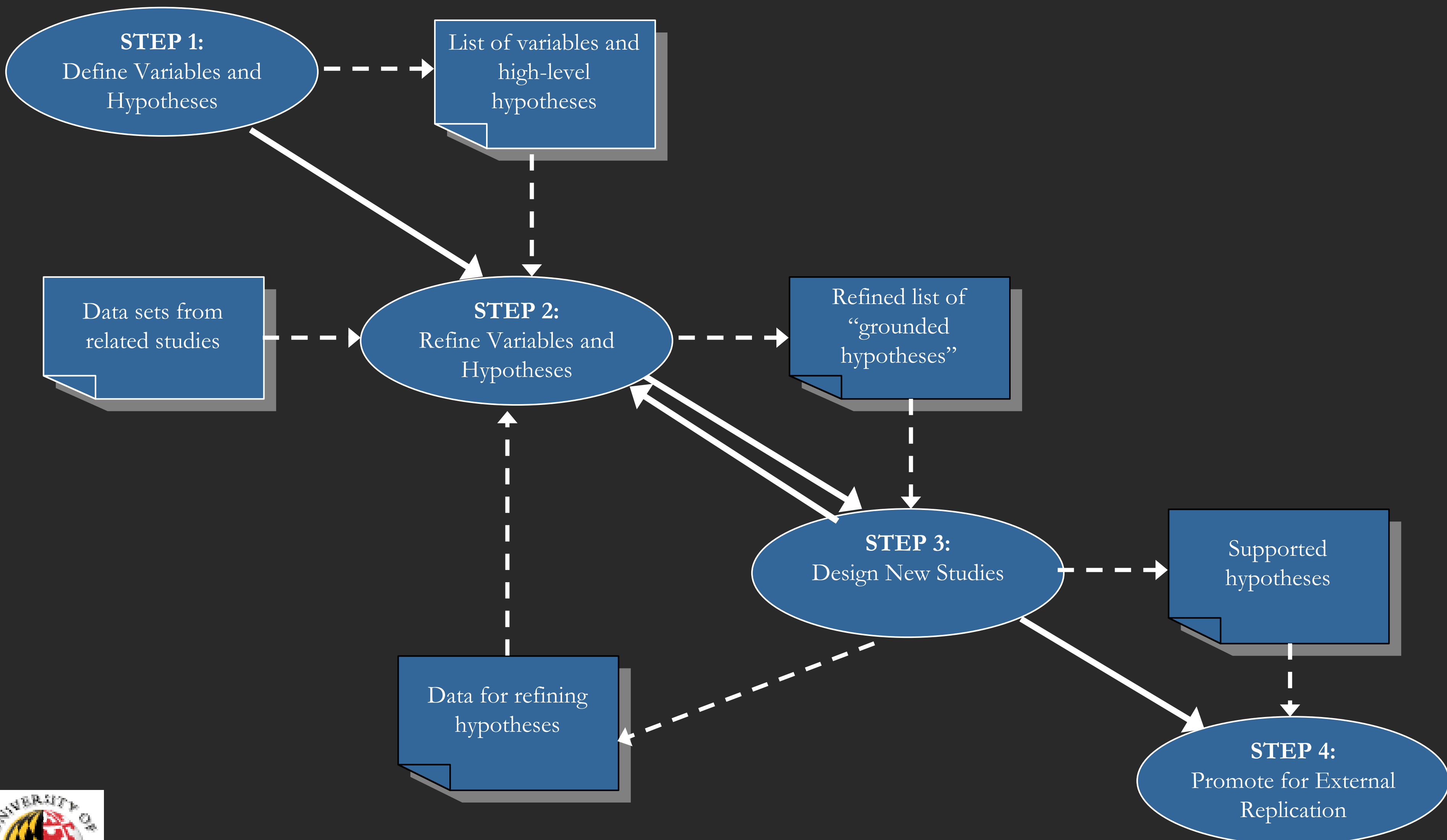
Step 1.1 – 1.3: Identify Variables

- Step 1.1
  - Siy96 – Inputs to an inspection important
- Step 1.2
  - Application Domain Knowledge (DK)
    - Dow94 at NASA; Fowler86 at AT&T; Parnas85 (Active Design Review)
  - Software Development Experience (DE)
    - Schneider92 (N-Fold Inspection); Cheng96 at Univ. New South Wales
  - Inspection Process Experience (PE)
    - Fusario97 IESE; Melo01 at Oracle
- Step 1.3
  - Training (T)
    - Collins89 and Pintrich96 from Education Literature
  - Motivation (M)
    - Alexander00 from Education Literature

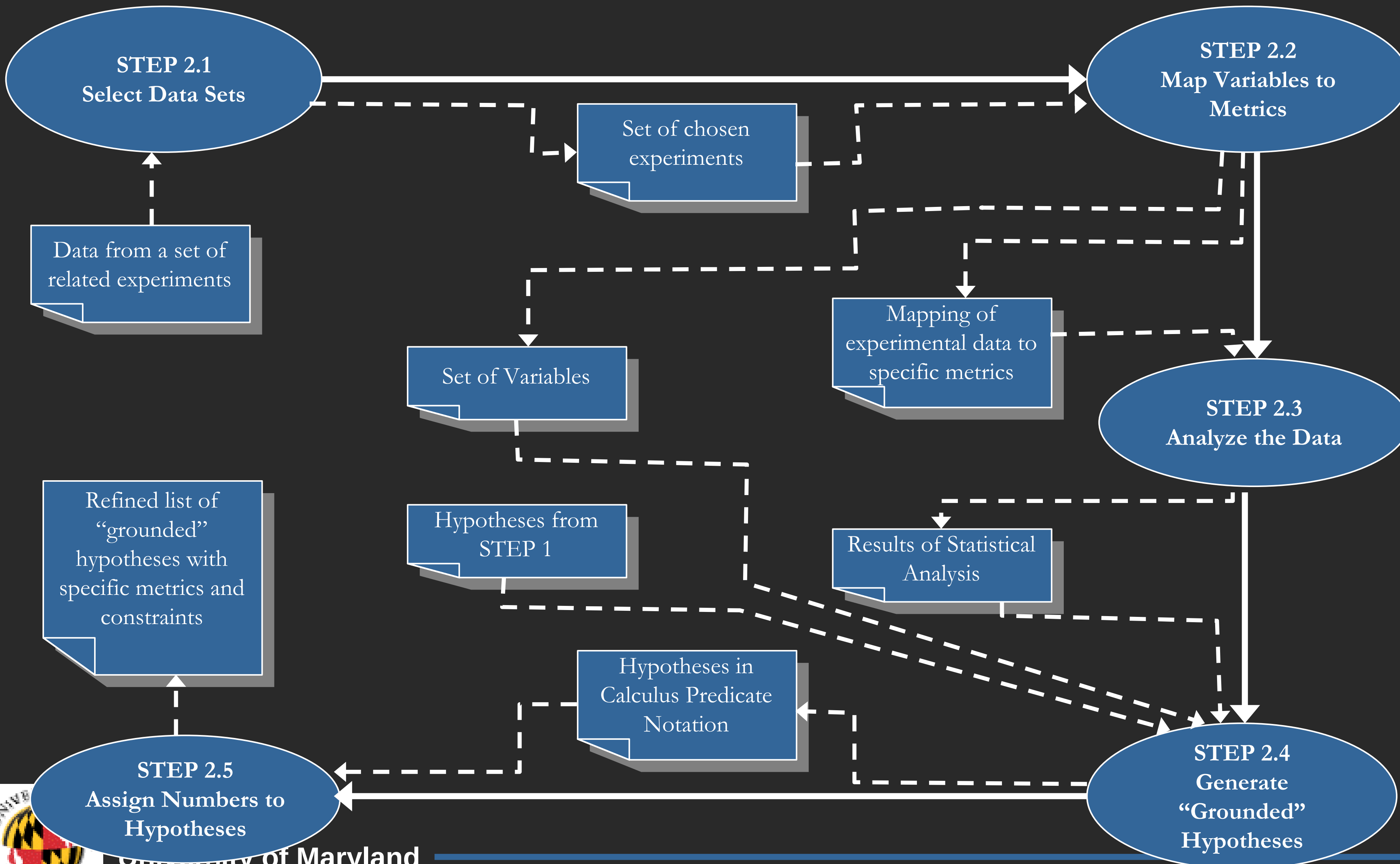
Step 1.4 – 1.5: Create Hypotheses

- Examples:
  - H.DK – Inspectors who have more domain knowledge will find more defects.
  - H.DE – Inspectors who have more software development experience will find more defects.
  - H.PE – Inspectors who are more experienced in the inspection process will find more defects.

Methodology



Step 2: Refine Variables and Hypotheses



STEP 2.1: Select Data Sets

- NASA 1994 & 1995
  - Two studies comparing PBR to Checklist at NASA in 1994 and 1995
- 735 Fall 1997
  - A study comparing PBR to Checklist (in isolation) in a graduate software engineering class at Maryland in 1997
- 438 Fall 1998
  - A study comparing PBR to Checklist (in a project) in an undergraduate software engineering class at Maryland in 1998
- 735 Fall 1999
  - A study to understand the steps of PBR and OORTs better in a graduate software engineering class at Maryland in 1999
- USC 2000 & 2001
  - A study to understand the use of PBR and OORTs in other development contexts at USC in 2000 and 2001

STEP 2.2 – Map Variables to Metrics

- Individual metrics collected in each study
  - Studies not designed with this research in mind
  - Metrics may not correspond directly to the variables of interest
  - Specific metrics should be mapped to variables and vice versa
  - E.g. Process Experience → Experience Reviewing Requirements and Software Development Experience → Requirements Experience

- If metrics cannot be mapped to existing variables, add new variables
  - E.g. Working Language Experience (WL) and Level of Process Specificity (PS)

STEP 2.3 – Analyze Data

STEP 2.4 – Generate Hypotheses

For a requirements inspection, inspectors who have more *experience working with requirements*, will **find more** defects than inspectors who have less *experience working with requirements*

			Requirements Experience (General)			
Experiment	Artifact	Technique	None/Class	Industry	p-value	
735 Fall 1997	ATM (Req)	Ad hoc	31.2% (29)	35.2% (36)	.307	√
	PGCS (Req)	PBR (U)	28% (13)	34.1% (9)	.327	√
		PBR (D,T)	28.3% (16)	29.6% (28)	.734	√
		PBR (U,D,T)	28.2% (29)	30.7% (37)	.418	√
	ATM & PGCS (Req)	Both	29.7% (58)	32.9% (73)	.196	√

STEP 2.3 – Analyze Data

STEP 2.4 – Generate Hypotheses

			Domain Knowledge			
Experiment	**Artifact**	**Technique**	**Non-expert**	**Expert**	**p-value**	
735 Fall 1999	LA & PG (Req)	PBR (U)	15.7% (6)	14.1% (2)	.841	X
		PBR (T)	22.2% (2)	27.3% (4)	.552	√
		PBR (U&T)	17.4% (8)	22.9% (6)	.325	√
	LA (Des)	OORTs	25.1% (7)	--	--	--
	PG (Des)	OORTs	18.6% (1)	11.6% (6)	.371	X
	LA & PG (Des)	OORTs	24.3% (8)	11.6% (6)	.002	X

For a requirements inspection, inspectors with more *domain knowledge* will **find more** defects.

For an OO design inspection, inspectors who have more *domain knowledge* will **find fewer** defects.

STEP 2.4 – Generate Grounded Hypotheses

			Experience Reviewing Requirements			
Experiment	**Artifact**	**Technique**	**None/Class**	**Industry**	**p-value**	
CMSC 435 Fall 1998	LA (Req)	Checklist	10.3% (17)	12.6% (3)	.722	√
		PBR (U)	10.3% (19)	12.4% (5)	.369	√
		Checklist & PBR (U)	10.3% (36)	12.5% (8)	.461	√
CMSC 735 Fall 1999	LA (Req)	PBR (U&T)	16.7% (7)	22.2% ([illegible])	.627	√
	PG (Req)	PBR (U&T)	23.4% (2)	22.7% (4)	[illegible]	X
	LA & PG (Req)	PBR (U)	16.2% (6)	12.7% (2)	.67	X
		PBR (T)	22.1% (3)	29.2% (3)	.369	√
		PBR (U&T)	18.2% (9)	22.6% (5)	.454	√
USC Fall 2000	--	PBR (T, U & D)	18.3% (20)	19.9% (10)	.895	√
USC Spring 2001	--	PBR (T, U & D)	26% (20)	18.9% (11)	.562	X

For a requirements inspection, inspectors who have more *experience reviewing requirements* will **find more** defects, when inspecting a requirements document which is unfamiliar or from an unfamiliar domain.

STEP 2.4 – Generate "Grounded" Hypotheses

- New Variables and Hypotheses Identified
  - Level of Process Specificity (PS)
    - H.PS – The level of process specificity necessary in an inspection is related to the level of experience of the inspector.
  - Working Language Experience (WL)
    - H.WL – Inspectors who are more proficient and comfortable in various aspects of the working language will find more defects.
      - For a requirements inspection, inspectors who are *native speakers* of the working language will **find more** defects <u>when the application domain is not familiar.</u>

STEP 2.4 – Generate "Grounded" Hypotheses

- Constrain the hypotheses where necessary
  - Add or remove so that hypotheses accurately describe available data
  - Limit the applicability of a hypothesis
    - The inspection process for which it holds
    - Characteristics of the inspector
    - The set of artifacts to which it applies

STEP 2.4 – Generate "Grounded" Hypotheses

- Predicate calculus notation for the hypotheses
  - (↑ Domain knowledge) ∧ (Requirements inspection) ⇒ (↑ Defects found)
  - (↑ Domain knowledge) ∧ (Design inspection) ⇒ (↓ Defects found)
  - (↑ Experience writing requirements) ∧ (Requirements inspection) ∧ (Ad hoc technique) ⇒ (↑ Defects found)
  - (↑Experience reviewing requirements) ∧ (Requirements inspection) ∧ (¬(document familiar) ∨ ¬(domain familiar)) ⇒ (↑ Defects found)
  - (↑ Software inspection experience) ∧ (Requirements inspection) ∧ (Same style of inspection) ⇒ (↑ Defects found)

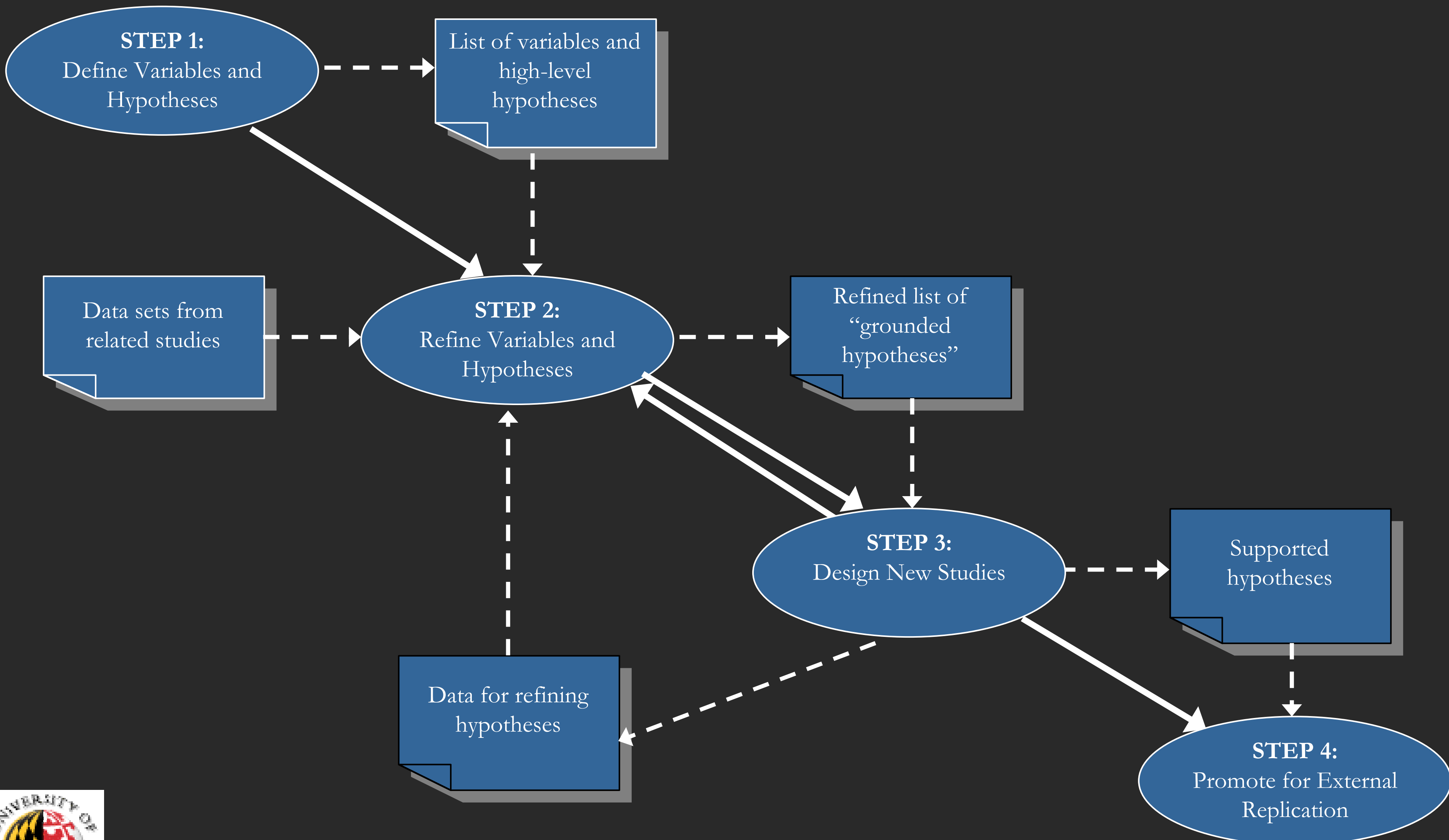
STEP 2: Second Iteration

- As new data becomes available iterate STEP 2
- Set of new studies
  - 4 Studies run in a Brazilian university/industry setting
  - Not specifically designed to test one of the hypotheses, but collected background metrics that allowed for analysis
- Hypotheses refined in the following ways:
  - Hypothesis supported
  - Adding or removing constraints
  - Adding new variables/hypotheses

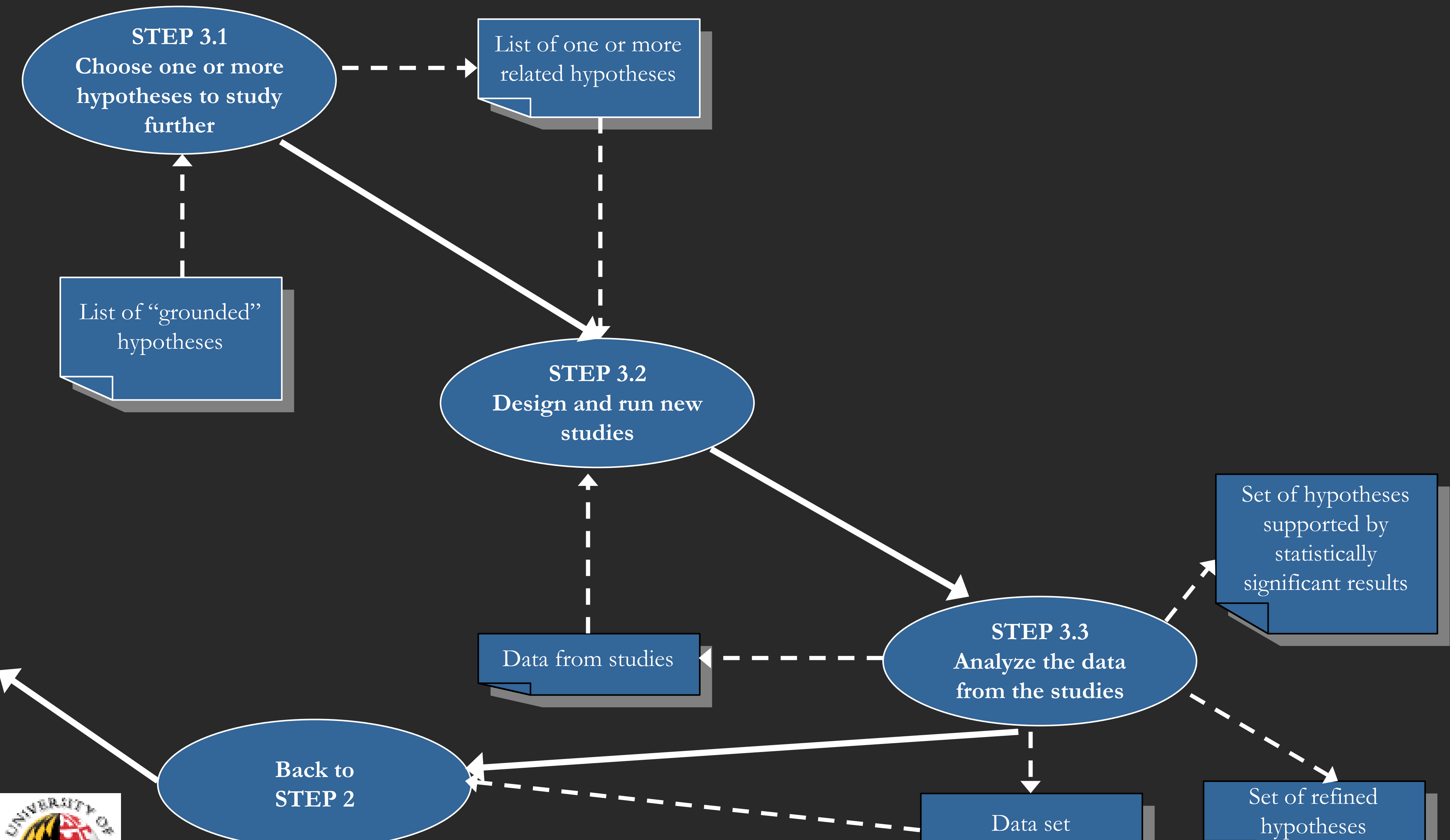
STEP 2: Second Iteration

- Refined hypotheses based on data:
  - Added a constraint on the technique:
    - ($\uparrow$ Domain knowledge) $\wedge$ (Requirements inspection) $\wedge$ (Non-procedural technique $\vee$ ad hoc technique) $\Rightarrow$ ($\uparrow$ Defects found)
  - Removed a constraint on the technique:
    - ($\uparrow$ Experience writing requirements) $\wedge$ (Requirements inspection) $\wedge$ (~~Non-procedural technique $\vee$ ad hoc technique~~) $\Rightarrow$ ($\uparrow$ Defects found)
  - Added constraint on inspector characteristics
    - ($\uparrow$ Experience reviewing requirements) $\wedge$ (Requirements inspection) $\wedge$ (Inspectors familiar with working language) $\wedge$ ($\neg$(Document familiar) $\vee$ $\neg$(Domain familiar)) $\Rightarrow$ ($\uparrow$ Defects found)
  - Added new hypotheses
    - ($\uparrow$ Experience in perspective) $\wedge$ (Requirements inspection) $\wedge$ (Low detail technique) $\Rightarrow$ ($\uparrow$ Defects found)

Methodology



STEP 3: Design New Studies



STEP 3.1 – Choose Hypotheses

- Two chosen for further study
  - **Hypothesis 1**: *Process Experience*
    - Inspectors who observe an inspection before performing one will find more defects than inspectors who do not observe an inspection before performing one
  - **Hypothesis 2**: *Level of Process Specificity*
    - To be effective, the level of detail in a technique must be tailored based on the inspector's experience. More experienced inspectors need less detail while less experienced inspectors need more detail

STEP 3.2 – Design New Studies

- Study 1 – *Process Experience*
  - **Background**
    - Open Questions about Process Experience
      - How much experience is necessary?
      - How can that experience be acquired?
    - Provide guidance on simplifying training and methods for building experience within an organization
  - **Goal**
    - Test the benefits of a specific type of process experience
  - **Hypothesis**
    - ($\uparrow$ Process experience) $\wedge$ (Requirements inspection) $\wedge$ (Same style of inspection) $\Rightarrow$ ($\uparrow$ Defects found)
      - Initially identified in first iteration of STEP 2 and supported in second iteration of STEP 2
    - (Observation of an inspection) $\wedge$ (Requirements inspection) $\wedge$ (Same style of inspection) $\Rightarrow$ ($\uparrow$ Defects found)

STEP 3.2 – Design New Studies

- Study 1 – *Process Experience*
  - **Procedure**
    - Background information collected via questionnaire
    - Subjects work in pairs
    - Subjects gain process experience by observing an inspection
  - **Data Collection**
    - Quantitative: defect lists, background questionnaire
    - Qualitative: observation, post-study questionnaire

STEP 3.3 Analyze the Data

Experience	Artifact	First Inspection	Second Inspection	p-value	
Low	PGCS	23.5% (3)	10.3% (4)	.23	X
	LA	12.5% (4)	31.5% (3)	.02	√
	PGCS & LA	17.2% (7)	19.4% ([illegible])	.738	√
High	PGCS	[illegible] (3)	[illegible] (3)	.643	√
	LA	18.5% ([illegible])	11.1% ([illegible])	[illegible]	X
	PGCS & LA	14.2% (6)	11.4% (6)	.484	X
High and Low	PGCS	16.7% (6)	10.9% (7)	[illegible]	X
	LA	15.1% (7)	21.3% (6)	.32	√
	PGCS & LA	15.8% (13)	15.7% (13)	.979	X

Results

Quantitative data: Some cases where subjects with experience found more defects than subjects without experience, but overall not much difference

Qualitative data: The observation process was helpful

Some New Hypotheses

(Observation of an inspection) ∧ (Requirements inspection) ∧ (Same style of inspection) ≠> (↑ Defects found)

(↑ Observation of an inspection) ∧ (Requirements inspection) ∧ (Same style of inspection) ⇒ (↑ Defects found)

(Performing an inspection) ∧ (Requirements inspection) ∧ (Same style of inspection) ⇒ (↑ Defects found)

STEP 3.2 – Design New Studies

- Study 2 – *Level of Process Specificity*
  - **Background**
    - **Open questions about Level of Process Specificity**
      - Does the level of process specificity matter in an inspection?
      - Is there an interaction between development experience and process specificity?
  - **Goal**
    - Understand the relationship between software development experience and required process detail
  - **Hypotheses**
    - $(\uparrow$ Experience in perspective$) \wedge ($Requirements inspection$) \wedge (\downarrow$ Technique detail$) \Rightarrow (\uparrow$ Defects found$)$
    - $(\downarrow$ Experience in perspective$) \wedge ($Requirements inspection$) \wedge (\uparrow$ Technique detail$) \Rightarrow (\uparrow$ Defects found$)$

STEP 3.2 – Design New Studies

- Study 2 – *Level of Process Specificity*
  - **Procedure**
    - Subjects worked individually
    - Assigned technique version based on experience
    - Two versions of PBR used
      - Low Detail Version
      - High Detail Version
  - **Data Collection**
    - Quantitative: defect lists, background questionnaire
    - Qualitative: post study questionnaire, class discussion

STEP 3.3 – Analyze the Data

	Group A	Group B	Group C
Perspective Experience	High	Low	Low
Process Detail	Low	High	Low
Defect Detection Rate	22.1%	20.6%	26.5%

Results

Quantitative Data: Little difference among subjects

Qualitative Data: Techniques seemed to contain the right amount of detail.

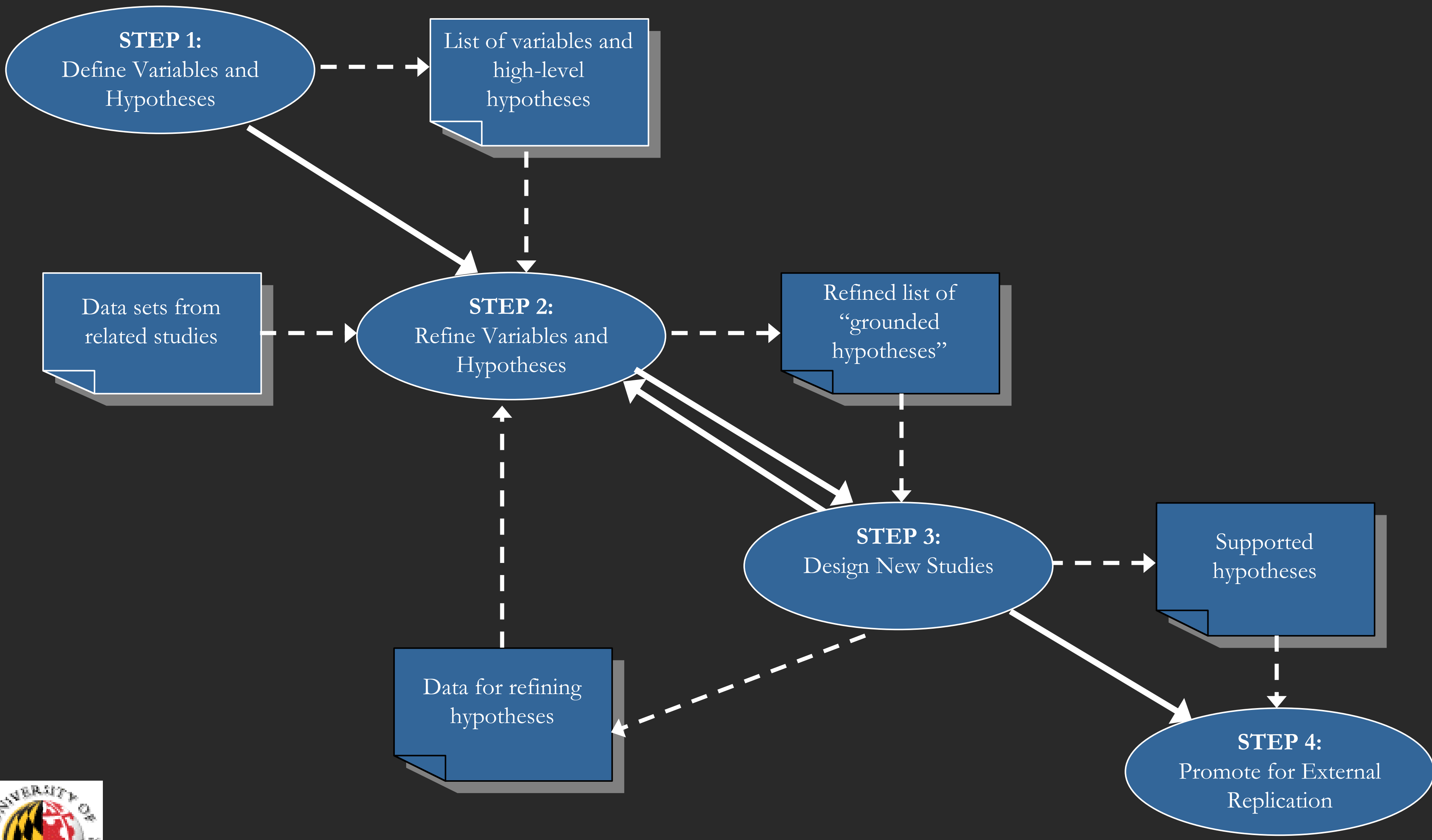
Artifacts not complex enough to see the benefits.

Two New Hypotheses

(↑ Experience in perspective) $\wedge$ (Requirements inspection) $\wedge$
(↓ Technique Detail) $\wedge$ (artifact of sufficient complexity)
$\Rightarrow$ (↑ Defects found)

(↓ Experience in perspective) $\wedge$ (Requirements inspection) $\wedge$
(↑ Technique detail) $\wedge$ (Artifact of sufficient complexity)
$\Rightarrow$ (↑ Defects found)

Methodology



Methodology:
Conclusions

- Showed that grounded theory is useful for conducting this type of research
- Was able to identify and refine a set of variables and hypotheses based on empirical data
- Identified and classified a series of constraints
- Represented hypotheses in predicate calculus to support future extensions on work

Methodology:
Impact of Results

- For Researchers:
  - A set of research questions and hypotheses to be investigated further (e.g.):
    - (Performing an inspection) $\wedge$ (Requirements inspection) $\wedge$ (Same style of inspection) $\Rightarrow$ ($\uparrow$ Defects found)
    - ($\uparrow$Domain knowledge) $\wedge$ (Design inspection) $\Rightarrow$ ($\downarrow$Defects found)

- For Practitioners:
  - Guidance to inspection planners concerning the composition of their inspection team
    - Testing experience seems to be important for inspectors
    - Reading techniques can neutralize the effects of different types of experience
    - Process detail must match inspectors’ experience

Methodology:
Contributions

- To Research Community:
  - A qualitative methodology for evolving hypotheses
  - A set of evolved hypotheses, supported by data
  - Ideas and hypotheses for future studies

- To Practitioner Community:
  - Advice for selection of inspectors
  - Advice on training
  - Advice on techniques to use

Methodology:
Limitations and Future Work

- Limitations
  - Most data drawn from students
  - Results based on studies designed for a different goal

- Future Directions for work
  - Augment methodology to:
    - Provide a “Level of support” for each hypothesis generated
    - Give more guidance on hypothesis selection
    - Focus on processes and artifacts as well as people
  - Conduct more studies
  - Broaden use of the methodology

Conclusions

- Empirical study is
  - An important tool for advance the state of a field
  - The missing link between conventional wisdom and supported facts

- Empirical methodology was useful in SE and now being used in HPC
  - Two fields differ
    - Types of hypotheses generated and studied
    - Availability of historical background data
    - Goals of the community

References

- [Alexander00] Alexander, P.A. "Toward a Model of Academic Development: Schooling and the Acquisition of Knowledge." *Educational Researcher* 29(2): 28-33,44. March 2000.
- [Cheng96] Cheng, B. and Jeffery, R. "Comparing Inspection Strategies for Software Requirements Specifications." In *Proceedings of the 1996 Australian Software Engineering Conference*, p. 203-211.
- [Collins89] Collins, A., Brown, J.S., and Newman, S.E. "Cognitive Apprenticeship: Teaching the Crafts of Reading Writing and Mathematics." In *Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser*, L.B. Resnik (ed.), Hillsdale, NJ: Erlbaum, 1989, p. 453-494.
- [Dow94] Dow, H.E. and Murphy, J.S. "Detailed Product Knowledge is not a Prerequisite for an Effective Formal Software Inspection." In *Proceedings of the 7th Software Engineering Process Group Meeting*, Boston, MA, May 1994.
- [Fowler86] Fowler, P.J. "In-process Inspections of Workproducts at AT&T." *AT&T Technical Journal*. 65(2): 102-112, March-April 1986.
- [Fusario97] Fusario, P., Lanubile, F., and Visaggio, G. "A Replicated Experiment to Asses Requirements Inspection Techniques." *Empirical Software Engineering: An International Journal* 2(1), 1997. p. 39-57.
- [Melo2001] Melo, W.; Shull, F. and Travassos, G.H., 2001. Software Review Guidelines. Technical Report ES-556/01. Systems Engineering and Computer Science Program. COPPE. Federal University of Rio de Janeiro. September. http://www.cos.ufrj.br/publicacoes/reltec/es55601.pdf
- [Parnas85] Parnas, D.L. and Weiss, D.M. "Active Design Reviews: Principles and Practice**."** ***Proceedings of the 8th International Conference on Software Engineering*****,** 1985, 132-136.
- [Pintrich96]Pintrich, P.R. and Schunk, D.H. *Motivation in Education: Theory Research and Practice*. "Chapter 5: Other Social Cognitive Processes." Englewood Cliffs: Prentice Hall, 1996. pp.153-197.
- [Schneider92] Schneider, G.M., Martin, J. and Tsai, W.T. "An Experimental study of Fault Detection in User Requirements Documents." *ACM Transactions on Software Engineering and Methodology* 1(2), April 1992. p. 188-204.