Doctoral research in CSE

What's the goal and how to get there?

Matt Dwyer
University of Nebraska – Lincoln

with lots of help from students and colleagues

What is research?

"creative work undertaken systematically to increase knowledge" (Frascati Manual)

Exploratory research

- identify and define a problem/question

Constructive research

- test theories and propose solutions to a problem/question

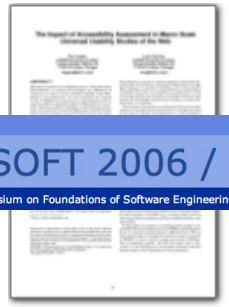
Empirical research

- test feasibility of a solution using empirical evidence

Why do research?

- To satisfy intellectual curiosity
- To better understand things
- To be at the forefront of an exciting, technical field
- To always be learning new things
- To make an impact on the world
- To get a job
- To keep a job (e.g., get tenure)
- …

The Path



ACM | SIGSOFT | FSE 14, 2006

ACM SIGSOFT 2006 / FSE 14

Fourteenth ACM SIGSOFT Symposium on Foundations of Software Engineering

Tasktop

eclipse

Atlassian

Microsoft Visual Studio

The Path

November 7-11 • 2010
Santa Fe·New Mexico·USA
2010 FSE-18
ACM SIGSOFT
Eighteenth International Symposium on the Foundations of Software Engineering

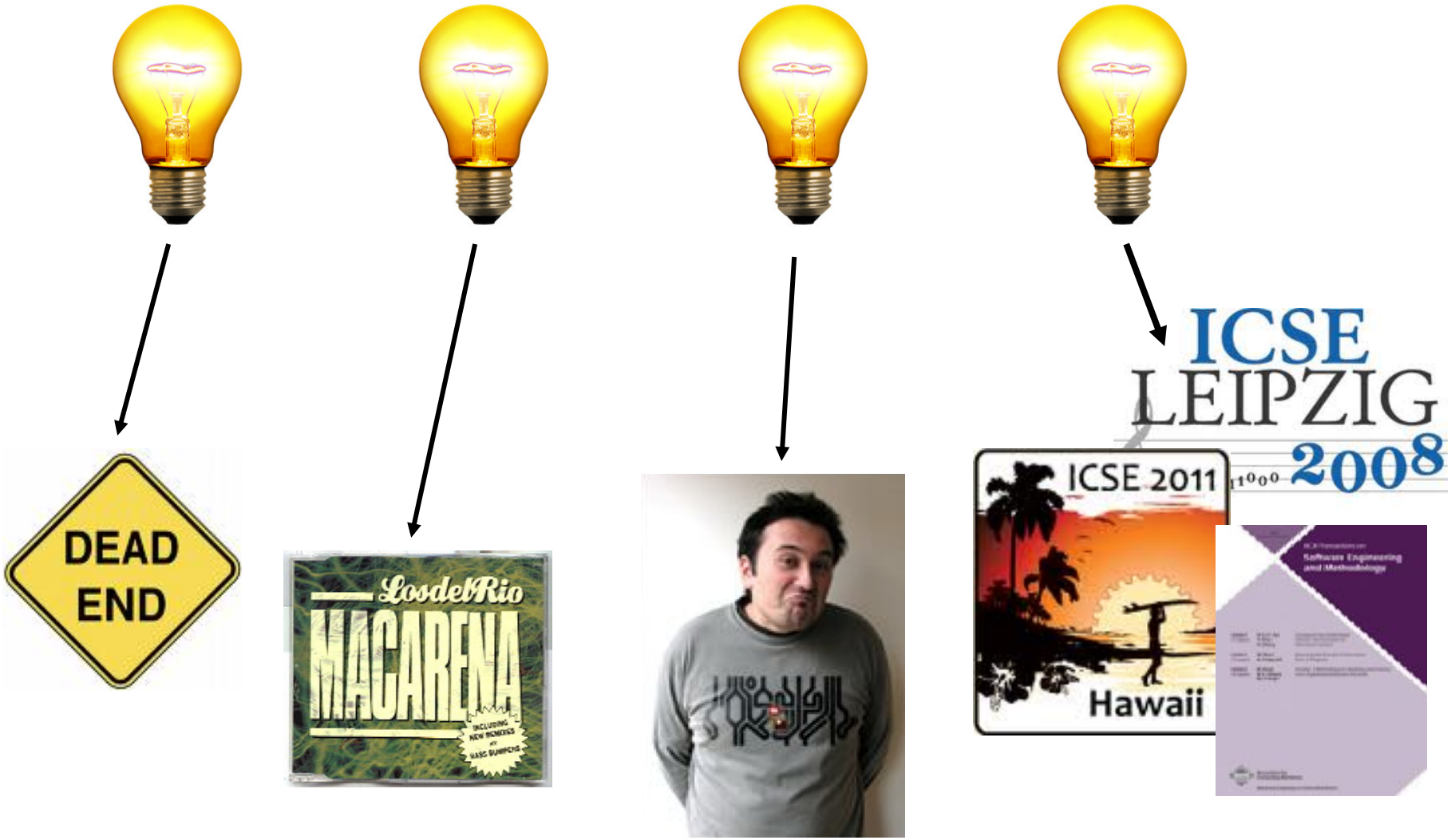
34th
International Conference on
Software Engineering®
2012

IEEE TRANSACTIONS ON
SOFTWARE
ENGINEERING

The Path


November 7-11 • 2010
Santa Fe·New Mexico·USA
2010 FSE-18
ACM SIGSOFT
Eighteenth International Symposium on the Foundations of Software Engineering
34th
International Conference on
Software Engineering®
2012
IEEE TRANSACTIONS ON
SOFTWARE
ENGINEERING

The Challenge



Focus on Ideas not Papers

For any problem/idea only 3 papers matter:

The First

introduces the problem/idea to the world

The Best

everyone looks to this paper to understand it

The Last

close the book on the problem/idea

Choosing a Research Problem

Problem should have potential impact

If you solve it someone will care …

- other researchers
- practitioners

Just because you can, doesn't mean you should

Google runtime verification ltl

Scholar About 5,680 results (0.07 sec)

Articles
Case law
My library

Any time
Since 2016
Since 2015
Since 2012
Custom range...

Sort by relevance
Sort by date

include patents
include citations

Create alert

Runtime verification for LTL and TLTL [PDF] uni-luebeck.de
A Bauer, M Leucker, C Schallhart - ACM Transactions on Software ..., 2011 - dl.acm.org
Abstract This article studies **runtime verification** of properties expressed either in lineartime temporal logic (**LTL**) or timed lineartime temporal logic (TLTL). It classifies **runtime verification** in identifying its distinguishing features to model checking and testing, ...
Cited by 290 Related articles All 20 versions Cite Save

Comparing **LTL** semantics for **runtime verification** [PDF] nicta.com.au
A Bauer, M Leucker, C Schallhart - Journal of Logic and ..., 2010 - Oxford Univ Press
Abstract When monitoring a system wrt a property defined in a temporal logic such as **LTL**, a major concern is to settle with an adequate interpretation of observable system events; that is, models of temporal logic formulae are usually infinite words of events, whereas at ...
Cited by 136 Related articles All 16 versions Cite Save

[HTML] A brief account of **runtime verification** [HTML] sciencedirect.com
M Leucker, C Schallhart - The Journal of Logic and Algebraic Programming, 2009 - Elsevier
... 126–138. [10]; A. Bauer, M. Leucker, C. Schallhart. **Runtime verification** for **LTL** and TLTL, Technical Report TUM-I0724, TU München, 2007. [11]; A. Bauer, M. Leucker, C. Schallhart. The good, the bad, and the ugly—but how ugly is ugly? ...
Cited by 358 Related articles All 11 versions Cite Save

[HTML] Temporal assertions using AspectJ [HTML] sciencedirect.com
V Stolz, E Bodden - Electronic Notes in Theoretical Computer Science, 2006 - Elsevier
... Abstract. We present a **runtime verification** framework for Java programs. Properties can be specified in Linear-time Temporal Logic (**LTL**) over AspectJ pointcuts. ... Keywords. **Runtime verification**; **LTL**; AspectJ; aspect-oriented programming; alternating automata. References. ...
Cited by 178 Related articles All 10 versions Cite Save

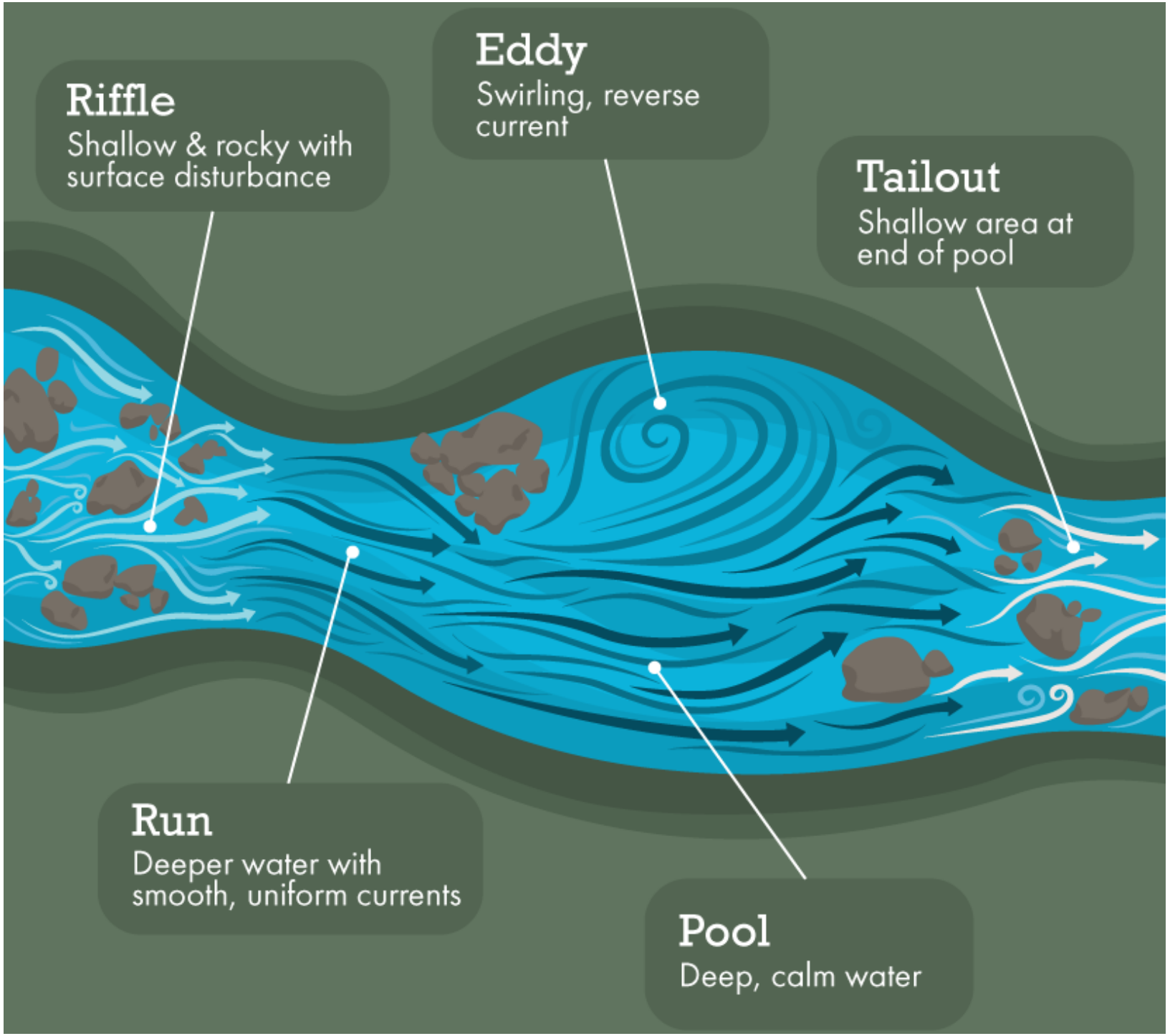
Rule-based **runtime verification** [PDF] nasa.gov
H Barringer, A Goldberg, K Havelund, K Sen - ... Workshop on **Verification**, ..., 2004 - Springer
... It is highly scalable. Several **runtime verification** systems have been developed, of which some were presented at three recent international workshops on **runtime verification** [1]. Linear temporal logic (**LTL**) [19] has been core to several of these attempts. ...
Cited by 342 Related articles All 30 versions Cite Save

[HTML] **Runtime verification** of timed **LTL** using disjunctive normalized equation systems [HTML] sciencedirect.com
KJ Kristoffersen, C Pedersen, HR Andersen - Electronic Notes in ..., 2003 - Elsevier
In this paper we present a new framework for **runtime verification** of properties of real time systems such as financial systems or backend databases. Such a systems has a semantics which resemples that of timed traces, namely a sequence of states where each state ...
Cited by 49 Related articles All 2 versions Cite Save

[HTML] **Runtime verification** of concurrent Haskell programs [HTML] sciencedirect.com
V Stolz, F Huch - Electronic Notes in Theoretical Computer Science, 2005 - Elsevier
... **LTL** formulas specifying assertions or other properties are **verified** at **runtime**. ... It is possible to dynamically add formulas at **runtime**, giving a degree of flexibility which is not available in static **verification** of source code. ...
Cited by 40 Related articles All 10 versions Cite Save

A lightweight **LTL runtime verification** tool for Java [PDF] psu.edu
E Bodden - Companion to the 19th annual ACM SIGPLAN ..., 2004 - dl.acm.org
Abstract **Runtime verification** is a special form of **runtime** testing, employing formal methods and languages. In this work, we utilize next-time free linear-time temporal logic (**LTL**\ textbackslash X) as formal framework. The discipline serves the purpose of asserting ...
Cited by 32 Related articles All 6 versions Cite Save

Runtime verification of LTL-based declarative process models [PDF] unibz.it


Riffle
Shallow & rocky with surface disturbance
Eddy
Swirling, reverse current
Tailout
Shallow area at end of pool
Run
Deeper water with smooth, uniform currents
Pool
Deep, calm water

Choosing a Research Problem

Problem should hold your interest

A good problem will occupy you for years

- Getting up to speed
- Developing initial results
- Spinning off several new threads

You should enjoy the techniques, methods, and challenges involved with the problem

Choosing a Research Problem

Problem should have depth

Good research addresses multiple dimensions

- Defining the problem
- Establishing the foundations of a class of solutions
- Realizing solutions with different tradeoffs
- Evaluating the cost-effectiveness of solutions

Look for problems that allow room to run

How to proceed?

Where to find good problems …

Question existing literature

- Does precision of points-to analysis matter?

Bring results from another area

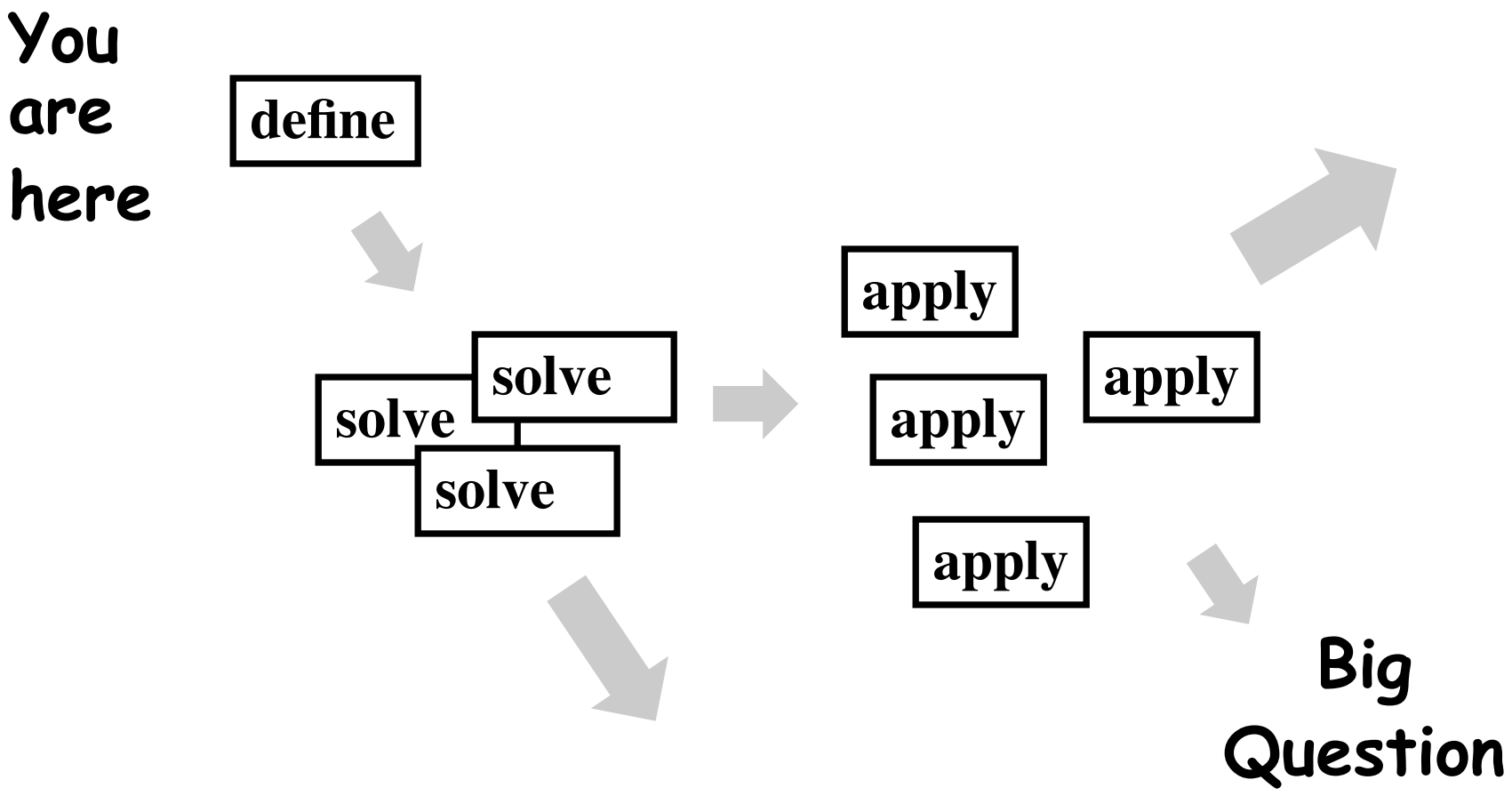
- constraint solving, search, data mining, …

Understand the experience of modern software development

- How do groups coordinate work?

How to proceed?

Think strategically, act tactically



How to proceed?

Think strategically, act tactically

Frame long-term questions (>5 years)

- e.g., longitudinal/differential analysis

Plan and develop a series of short-term efforts (3-12 months)

- e.g., define differential symbolic execution (FSE'08), implement tools ('10), scale tools (PLDI'11), apply in novel ways ('12)

How to proceed?

Think strategically, act tactically

Frame long-term questions (>5 years)

- e.g., can model checking be applied to software? (in 1996)

Plan and develop a series of short-term efforts (3-12 months)

- e.g., specifying properties (ICSE'99), Bandera (ICSE'00), abstraction (ICSE'01), lots of other steps by us and others

How to proceed?

Know the literature

What literature?

- SE and allied fields (PL,IR,DB,CAV,…)

How much should you read?

- A lot (>75 papers a year)

Keep up since things move quickly

- Scan good meetings, make a list, read on your commute/in bed/etc.
- Follow other researchers, visit their web-sites

How to proceed?

Critically examine previous approaches

- What assumptions are made?
- How scalable/practical is the approach?
- Has the approach been evaluated well?
- Are there missing connections with other approaches?
- If this is a definition are there good solutions?
- If this is a solution are their valuable applications?

How to proceed?

Know what it means to "solve" a problem or "validate" a solution

Find some models in the literature

- e.g., Tom Henzinger's work (Interface Automata, CPA, Blast, …)

Think about what you like/dislike in models

- e.g., pros: clarity and precision, cons: rarely evaluated at scale

How to proceed?

Reflect on your own research

How do you compare to models?
What could you improve?
Are you making progress towards strategic goals?

Do this regularly

- Beginning of every semester, end of year
- For yourself and your team

How to proceed?

Plan to be productive

What are your research products?

- Software, data sets, prototypes, papers, …

How will you disseminate them?

- Passive dissemination is limited
- Be proactive, e.g., submit to the best venues

Assess your productivity

- Are you publishing in the best places/regularly?
- If not, is it you, your collaborators, …

Research logistics

Forcing functions are your friend

Deadline-driven work isn't necessarily bad

- Aim for paper deadlines
- Commit to giving talks
- Commit to collaborating

Deadlines are only bad if you do poor work

- pull back if you aren't happy with results

Research logistics

Working with your advisor

Prove that you are serious and reliable

- Set milestones and expectations, meet them!

Advisor is an accomplished researcher, learn from them

- Advisor leads first effort
- Collaborate on next effort

You MUST learn to drive your research

- Lead your advisor
- Leave them behind!!

Research logistics

Employee, Apprentice, Colleague, …

Initially an employee

- Assigned work, expected to finish it promptly and well

Ultimately a colleague

- You collaborate with your advisor as a peer

The transition is up to you

- Be reliable (meet deadlines, high quality work)
- Be independent (bring new ideas to the collaboration)
- Staying an "employee" is cause for concern

Research logistics

Develop a research style

Dreamer: Internal idea development and "imagining"

Explorer: Rapid prototype to gain understanding before diving in

Deep thinker: One thing at a time, works to completion

Juggler: Likes to do different things, context-switches easily

Planner: Lay out a series of steps and follow them

Most good researchers are a hybrid of these

Invest for Impact

What is impact?

What do you have to invest?

How to invest?

What is Impact?

The extent to which the knowledge you create influences others

Who is influenced

- researchers, practitioners, students, the public at large

Kinds of influence

- New questions, concepts, techniques, … (build mindshare)
- Demonstration of value (build business case)

What to invest?

Invest time to gain knowledge

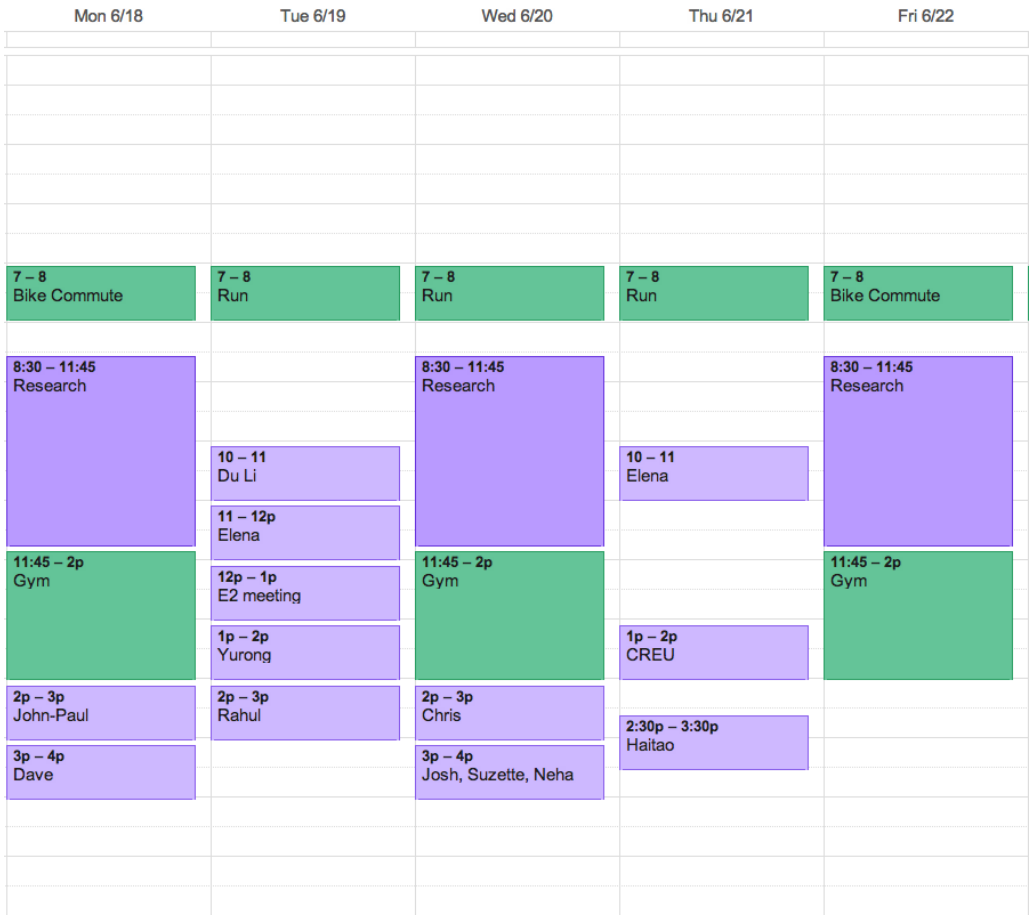
Your own time, your team's time

You will have many competing demands on a limited time budget

- Classwork, Qualifying Exam Prep, …
- Reading research papers
- Writing code, Running experiments
- Thinking
- Life

Research logistics

Establish uninterruptible blocks of 'research thinking time' in your schedule



How to invest?

Balance risk-reward and timescales

Invest in some "sure things"

- e.g., the next step in your current line of research
- A "brick"

Invest for high-reward

- e.g., build a connection between two areas (CIT and SAT)

Invest for short and long-term

- Get a paper out the door
- Develop the next big idea

How to invest?

Exploit the compounding effect

Acquire lasting knowledge

- Each increment will build on previous one

Learn something new frequently

- Model Checking (1996), Temporal logics (1997), Program Slicing (1999), Abstract Interpretation (2001), Structural Operational Semantics (2003), Runtime monitoring (2006), SAT Modulo Theories (2008), Markov Decision Processes (2010), Model Counting (2011), Probabilistic Modeling (2013)

How to invest?

Avoid diminishing returns

You are good at some things

- Leverage these strengths
- Keep them up to date

You are weak at some things

- Be honest with yourself about this
- This is where the biggest improvement can happen

Three Keys

Lead from strength

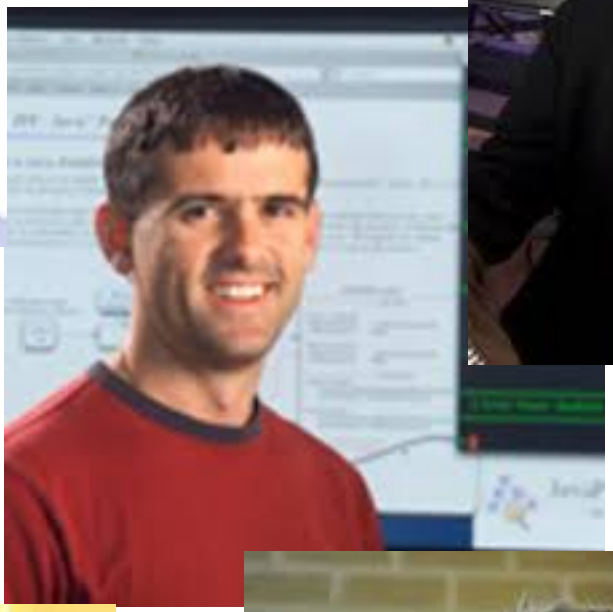
- But strengthen weaknesses over time

Ask big questions

- But break them up into smaller questions
- Plan on multiple timescales (1 month, 6 month, 1 year, 5 years)
- Hold yourself accountable for progress

Invest in yourself

- Grad school is once in a lifetime, commit to it
- Every year pick something new to learn
- Every year pick a research problem to lead







NSF