Tactical Decisionmaking in Research

25 March 2009

Research Methods for
Empirical Computer Science
CMPSCI 691DD

Project Report	Date
Report 3: Assess. of Current Knowledge	4/1/09
Report 4: Research Proposal	4/8/09
Research Proposal Reviews	4/15/09
Report 5: Experimental Design	4/22/09
Experimental Design Reviews	4/29/09
Report 6: Experimental Results	5/13/09
Report 7: Final Report	5/20/09

Note: Seven of the remaining 14 classes have no response paper due

I.I. Rabi

The ‘witz’

“Once, during the time in which I was studying the properties of pions at the Columbia-Nevis cyclotron, I told Rabi of the latest of these experiments. His rather unappreciative response was the question, ‘But what's the witz?’ ‘Witz’ is Yiddish for ‘cleverness.‘ Rabi's point was clear: To be interesting and fun, it was not enough for him that the physics goal of an experiment be important, he also wanted the experimental technique to show originality.”

— Jack Steinberger

J. Steinberger (2005). *Learning about Particles: 50 Privileged Years*. Springer.

Approaches to evaluating hypotheses

Approach	Ideal to use when system and environment* under study are...
Experiments	Complex, Poorly understood, and Implemented
Simulations	Complex, Well understood, and Unimplemented
Theorems	Relatively simple, Well understood, and Unimplemented

*or portions or aspects of the system and environment

Which approach should I use?

- This is usually the wrong question.
- The best answers are provided by...
  - ...*converging lines of evidence*
  - ...from multiple approaches (experiments, simulations, *and* theorems)...
  - ...and multiple versions of each approach.
- You don't have to provide all approaches (other investigators' work may supply them)...
- ...but you should be able to apply the approaches most relevant to your question

Experimental platforms vs. prototypes

- Experimental platform
  - A flexible implementation for the purpose of running specific experiments and exploring design space
  - Designed based on *key research questions*
  - Low implementation cost (though depends on the range of experiments it is intended to support)
- Prototype
  - An alpha or beta version of one specific design intended to be fully functional in a range of operational environments
  - Designed based on *key user needs*
  - High implementation cost
- A common mistake is building a prototype when an experimental platform would be more useful and less costly

Tactics for research programming

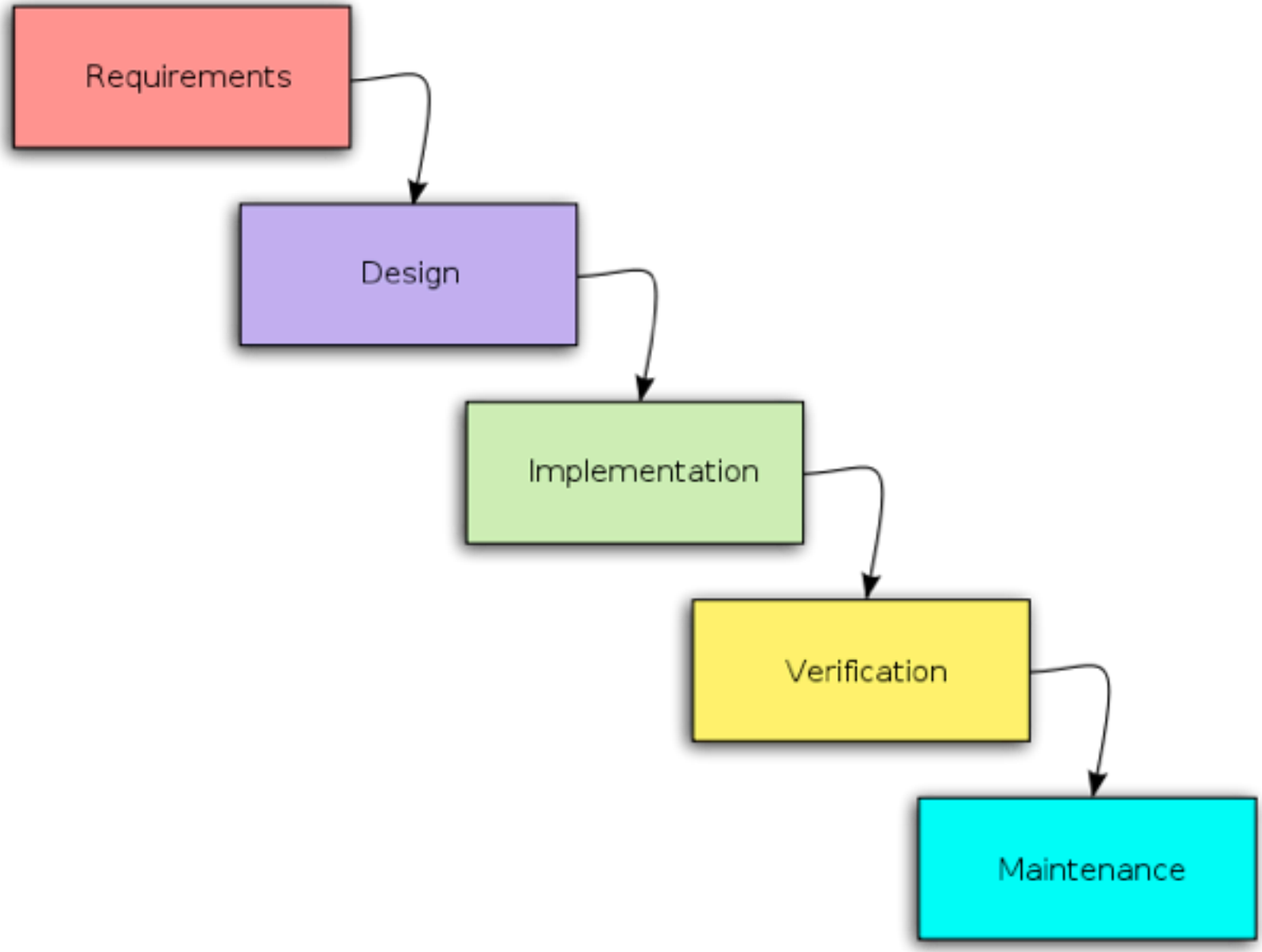
- Don't do it
  - Don't write new code if you don't have to
  - Use standard implementations whenever possible
- Emphasize *modularity*
  - For exploratory work, wire together existing components
  - For longer-term work, write modules that can be combined with scripts or simple high-level methods
- Code with the expectation of frequent *refactoring*
- Write *unit tests* for all core code
- *Document* key implementation choices

Tactics for managing system complexity

- Decompose a system into modular components
- Formally specify intermediate data structures produced by each component
- Specify data structures with *externally verifiable meaning* so that they can be evaluated independently
  - *Language models* in information retrieval
  - *Model structure and conditional probability distributions* in Bayesian network learning

Research processes

- Most computer scientists are familiar with one or more *software development processes*
- These processes are intended to direct the day-to-day activities of software developers in order to maximize the probability of a successful project
- Understanding of software development processes has improved substantially in the past several decades...


Requirements
Design
Implementation
Verification
Maintenance


COMMUNICATION
SIMPLICITY
On-Site Customer
Open Workspace
Acceptance Tests
User Stories
Test-Driven Development
Collective Ownership
Coding Standard
One Team
Pair Programming
Refactoring
Retrospective
Continuous Integration
Sustainable Pace
Simple Design
Iterations
Release Plan
Metaphor
Small Releases
FEEDBACK
COURAGE

"Agile research"

- Many of the lessons of "agile software development" apply to research
- Changes to research questions ("requirements") are common as understanding of those questions evolves
- Constructing large infrastructures without using them to examine research questions ("testing against requirements") can waste resources
- The "waterfall model" works poorly in research; Frequent iterations with short cycle times work better
- Courage, feedback, communication, and simplicity are just as needed in research as in software development