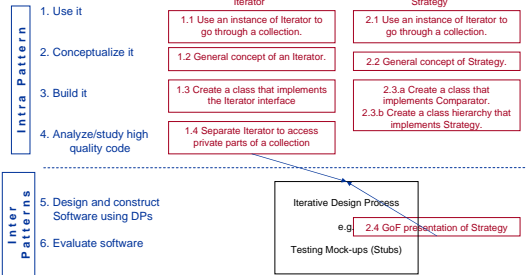


# "Killer Examples" for Design Patterns and Objects First Workshops

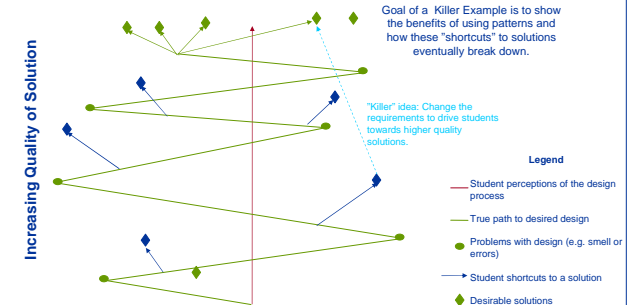
<http://www.cse.buffalo.edu/faculty/alphonc/KillerExamples>

OOPSLA 2005, San Diego • OOPSLA 2004, Vancouver • OOPSLA 2003, Anaheim • OOPSLA 2002, Seattle

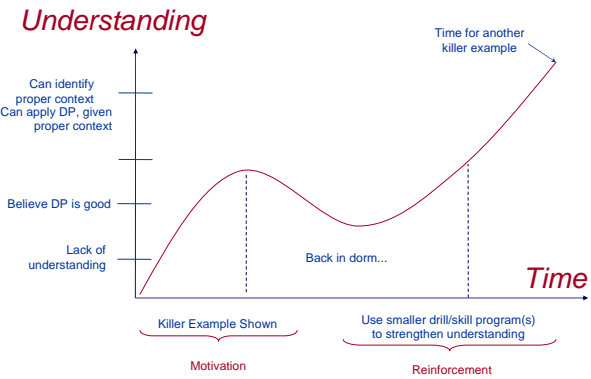
## Pedagogical Process for Teaching DPs



Carl Alphonc University at Buffalo alphonc@cse.buffalo.edu	Michael E. Caspersen University of Aarhus mec@dimi.au.dk	Adrienne Decker University at Buffalo adrienne@cse.buffalo.edu	Martha Kosa Tennessee Technological University MKKosa@tntech.edu	Stephen Wong Rice University swong@cs.rice.edu
<b>Killer Example</b> The Jargon File defines a "killer app" as an "application that actually makes a sustaining market for a promising but under-utilized technology." In the same vein, we take a "killer example" to be one which provides clear and compelling motivation for some concept. Killer Examples must be an integral component of a larger, cohesive OO curriculum. They do not exist in a vacuum, but rather as part of a deliberate pedagogical progression that drives from motivation to comprehension.				
A workshop to share ideas, specifically "killer examples", for <b>Fundamental Object-Oriented (OO) principles and Design Patterns</b> primarily in an objects-first CS1-CS2 sequence.				
<b>Motivation for Teaching Design Patterns</b> We want a systematic way to solve complex problems (need solutions that scale up). Design patterns support the building of correct, robust, flexible & extensible software in an efficient manner (time & \$). Important underlying principles which allow us to reach our goals: Abstraction Invariant/variant decoupling (commonality/variability analysis) Parameterization Extreme encapsulation (high abstraction; program to invariant behavior; decoupling to manage complexity)				
Design Patterns can be used to illustrate fundamental Computer Science principles. Design Patterns change and shape the way we look at problems.				



## Caspersen's Curve



## Killer Examples

<b>2002 Workshop</b> <b>OOPSLA 2002</b>  Seattle Java Power Tools R. Rasala, V. K. Prout & J. Raab (Northeastern)	Composition Framework D. Skrien (Colby College) Configuration Puzzles J. Heliotis and S. Marshall (RIT) Developing an Elevator Control System C. Nevison (Colgate) and B. Wells (South Fork High School) Kaleidoscope M. R. Wick (University of Wisconsin—Eau Claire) Properties of a "Killer Example" S. Sandall (Swiss Federal Institute of Technology)	A simple calculator for novice learning J. Bergin (Face University) Interactive Program Guide A. Sterkin (NDS Technologies) The Need for Killer Example for Object-Oriented Frameworks M. Caspersen & H. Christensen (University of Aarhus)	<b>2003 Workshop</b>  Foundation for Object-Oriented Graphics R. Rasala (Northeastern University)
<b>2004 Workshop</b> Generic Data Access in Microsoft .NET Joe Hummel (Lake Forest College)	 Applying the Extension Object Pattern to the Software Communication Architecture D. Parisicotti and B. Trask (SDR Products)	 Presentation Application ("PowerPoint") S. Stuurman (Open University, The Netherlands) and G. Florijn (SERC)	<b>2005 WORKSHOP</b> Design Patterns in JDK Collections C. Bouamalay (SBC Services) Killer Lab: Flow Simulation and Lead Poisoning Study J. Heliotis and G. Lutzler (RIT)
	 Message-Oriented Middleware Cache Pattern – A Pattern in a SOA Environment F. Yan, R. Fang and Z. Tian (IBM China Research Lab)	Using Design Patterns to Help Test Your Classes & Functions B. Trask, A. Roman and V. Bhanot (PRISMTECH)	

## Wong's Mapping

Underlying Principles

