NSF

USC CSE

University of Maryland 1856

CeBASE NSF Center for Empirically Based Software Engineering

University of Maryland • Fraunhofer Center-MD • University of Southern California • University of Nebraska • Mississippi State University

http://cebase.org

Mississippi State University

Nebraska
UNIVERSITY OF NEBRASKA-LINCOLN

Fraunhofer USA Center for Experimental Software Engineering Maryland

CeBASE

Outline

- Motivation
- Center Vision and Approach
- Center Organization
- Track Record
- Examples of Existing Empirical Results
- Collaborations
- Expected Benefits

CeBASE

Motivation for the Center

- **Software too fragile, unpredictable**
  - **(Presidential Commission Report)**
- **“No-surprise” software development**
  - **(NSF Workshop Report)**

CeBASE

Motivation for the Center

- **Software development teams need to understand the right models and techniques to support their projects. For example:**
  - **When are peer reviews more effective than functional testing?**
  - **When should you use a procedural approach to code reviewing?**
  - **How should you tailor a lifecycle model for your environment?**
- **Too often, such decisions are based on anecdote, hearsay, or hype**
- **Developers are often surprised to learn that 25 years of empirical software engineering research has provided usable answers to these and numerous related questions.**

CeBASE

CeBASE Vision and Approach

- **Goal: An empirically based software development process**
  - covering high-level lifecycle models to low-level techniques
  - necessary step toward a scientific foundation for software engineering in which the effects of development decisions are well understood
  - we can’t improve without a empirical base

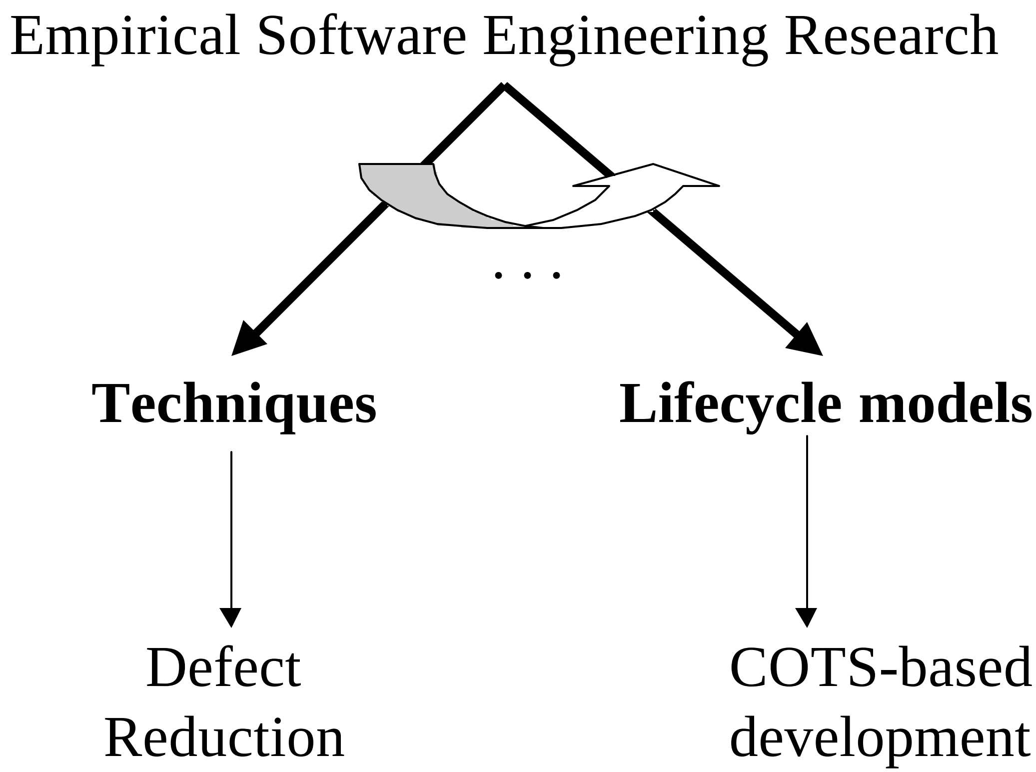
- **A first step is an empirical experience base**
  - validated guidelines for selecting techniques and models
  - ever-evolving with empirical evidence to help us
  - identify what affects cost, reliability, schedule,...

- **To achieve this we are**
  - Integrating existing data and models
  - Initially focusing on new results in two high-leverage areas...

CeBASE Approach

Investigates a spectrum of activities...

Initially we will focus on...



CeBASE

Center Organization

- **Research center sponsored by**
  - NSF Information Technology Research Program
- **Co-Directors**
  - Victor Basili (UMD), Barry Boehm (USC)
- **Co-PI's**
  - Marvin Zelkowitz (UMD), Rayford Vaughn (MSU), Forrest Shull (FC-MD), Dan Port (USC), Ann Majchrzak (USC), Scott Henninger (UNL)
- **Initial 2-year funding: $2.4 M**

CeBASE

Track Record of the PI’s

- **UMD and FC-MD**
  - 25 years of strong large-project research results
  - framework for empirical research widely adopted (GQM, EF, …)
  - leading to improved software engineering at NASA/Goddard Space Flight Center, CSC, DaimlerChrysler, Motorola…
- **USC-CSE**
  - Process frameworks and predictive models widely adopted
    - COCOMO, Spiral model, risk management framework, ...
  - Network of collaborators: FAA, US Army Research Labs, Hughes, Rational, ...

CeBASE

Software Engineering Laboratory Experiences
Observation, Feedback, Learning, Packaging

CeBASE represents a synthesis of much of the software engineering research at the SEL over the past 25 years

In the SEL, learning was based upon

- Observation
- Feedback loops
- Lessons learned packaged into the process, product and organization

Used the SEL as a laboratory to build models, test hypotheses

Developed technologies, methods, and theories to solve problems

Identified the parameters that made processes effective locally

Kept the business going with an aim at improvement, learning

CeBASE

SEL Activities Resulted in

Continuous Improvement in the FDD

Decreased **Development Defect rates** by
75% (1987 - 1991) **37%** (1991 - 1995)
Reduced **Cost** by
55% (1987 - 1991) **42%** (1991 - 1995)
Improved **Reuse** by
300% (1987 - 1991) **8%** (1991 - 1995)
Increased **Functionality** five-fold (1976 - 1992)

CSC

officially assessed as CMM level 5 and ISO certified (1998), starting with SEL organizational elements and activities

Fraunhofer Center

for Experimental Software Engineering - Maryland created 1998

CeBaSE

Center for Empirically-Based Software Engineering created 2000

CeBASE

SEL-Related Studies from FDD to ISC

COTS Studies

- COTS development process
- COTS Selection Model
- Architectural Mismatch Model
- COTS Integration / Cost estimation

Reading Techniques

- Requirements
- Object Oriented Design
- RFP

Experience Management System

- Experience Capture Methodology
- Experience Base tool

Current Empirical Research at USC: COCOMO - Constructive Cost Model

- **COCOMO81 has been the world's most widely-used software cost model**
  - **Over 15 commercial products**
- **COCOMO II book and CD (Prentice Hall, 2000)**
  - **Focused on new software processes and products**
  - **4 commercial products to date**
- **Several emerging extensions**
  - **COCOTS (COTS integration cost and schedule)**
  - **CORADMO (Rapid Application Development)**
  - **COQUALMO (Delivered defect density)**
  - **COPSEMO (Phase and activity distribution)**
  - **COPROMO (Productivity strategy assessment)**

Current Empirical Research at USC: MBASE - Model-Based (System) Architecting and Software Engineering

- **Extends spiral model for integrated product and process development**
  - **Balances discipline and flexibility**
  - **Stakeholder win-win approach**
  - **Life cycle anchor point milestones**
- **Avoids model clashes among stakeholder success, process, product, and property models**
  - **Major source of project failures and overruns**
- **Compatible with Integrated Capability Maturity Model (CMMI) and Rational Unified Process (RUP)**
- **Supported by electronic process guide, commercial tools**
  - **Easy Win Win requirements negotiation, Rational toolset**
- **Annually improved by empirical analysis of over 100 MBASE projects.**

CeBASE

Examples of Useful Empirical Results
Technique Selection Guidance

"Under specified conditions, …"

- **Peer reviews are more effective than functional testing for faults of omission and incorrect specification (UMD, USC)**
- **Functional testing is more effective than reviews for faults concerning numerical approximations and control flow (UMD, USC)**

CeBASE

Examples of Useful Empirical Results

Technique Definition Guidance

"Under specified conditions, …"

- **For a reviewer with an average experience level, a procedural approach to defect detection is more effective than a less procedural one. (UMD)**
- **Procedural inspections, based upon specific goals, will find defects related to those goals, so inspections can be customized. (UMD)**
- **Readers of a software artifact are more effective in uncovering defects when each uses a different and specific focus. (UMD)**

CeBASE

Examples of Useful Empirical Results
Lifecycle Selection Guidance

Lifecycle Selection Guidance

- **The sequential waterfall model is suitable if and only if**
  - **The requirements are knowable in advance,**
  - **The requirements have no unresolved, high-risk implications,**
  - **The requirements satisfy all the key stakeholders' expectations,**
  - **A viable architecture for implementing the requirements is known,**
  - **The requirements will be stable during development,**
  - **There is enough calendar time to proceed sequentially. (USC)**
- **The evolutionary development model is suitable if and only if**
  - **The initial release is good enough to keep the key stakeholders involved,**
  - **The architecture is scalable to accommodate needed system growth,**
  - **The operational user organizations can adapt to the pace of evolution,**
  - **The evolution dimensions are compatible with legacy system replacement,**
  - **appropriate management, financial, and incentive structures are in place. (USC)**

CeBASE

Industry Benefits

- **To advance software development, industry needs**
  - **a basis for choosing and customizing development approaches**

- **We will support industry by developing**
  - ***an understanding of defects (and a means to minimize them) grounded in careful empirical analysis instead of folklore***
  - ***empirical metrics and predictive models for process selection and project monitoring***

CeBASE

Research Benefits

- **To advance software engineering research, we must**
  - **identify and solve significant software development problems**
  - **validate the solutions**

- **We will support SE researchers to**
  - ***engage in collaborations with industry***
  - ***enable integration of results for more robust conclusions***
  - ***evaluate, refine, and extend results and methods***
  - ***package and disseminate results via educational materials and activities***

CeBASE

Educational Benefits

- **To advance software engineering education, we must**
  - teach high-impact methods
  - offer courses with relevant and timely results
  - give students experience with realistic artifacts
  - educate a stronger community of empirical researchers

- **We will support SE educators by**
  - ***providing material for training students on how to select and tailor, not just apply, the right SE methods and tools***
  - ***providing realistic artifacts as teaching materials***
  - ***designing SE educational techniques supporting experimentation***

Long-term Goal

- We are looking for collaborations with
  - development projects and support groups
  - research organizations
  - educational institutions
- To help
  - build, evaluate, and share the empirical experience base
  - build and validate design principles
- Evolve the software engineering discipline from fad-based to
  - empirically-based
  - scientifically-based
  - engineering-based